Aviation Forecast Preparation System

System and User Guide

AvnFPS Team, Product Generation Branch, Meteorological Development Laboratory

Aviation Forecast Preparation System: System and User Guide

by AvnFPS Team, Product Generation Branch

Published 1 August 2006

Table of Contents

Introduction ix
1. System Administration Manual 1
    1. System Overview 2
        1.1. Name and Event Server 4
        1.2. Data Ingest Server 5
        1.3. Data Request Server 6
        1.4. Transmission Server 7
        1.5. Client Programs 9
    2. Configuring AvnFPS 10
        2.1. Files in `etc` 11
        2.2. TAF configuration files 18
        2.3. TWEB configuration files 24
        2.4. X resources configuration files 25
        2.5. Climatological Data 27
    3. Starting and Stopping AvnFPS 29
        3.1. Starting AvnFPS Servers 30
        3.2. Stopping AvnFPS Servers 32
    4. Avnsetup 33
        4.1. Editing Monitoring Rules 34
        4.2. Editing TAF Site Information 37
        4.3. Editing TAF Site Templates 39
        4.4. Editing TAF Product Definition 40
        4.5. Editing TWEB Route Information 42
        4.6. Editing TWEB Route Template 44
        4.7. Editing TWEB Product Definition 45
        4.8. Creating Triggers 47
        4.9. Using Text Editor 48
    5. Configuring IFPS to send gridded data to AvnFPS 50
    6. Logging 51
    7. Troubleshooting 53
        7.1. Overall Server Health 54
        7.2. Log Files 56
        7.3. Data Ingest 57
        7.4. Climatology Files 58
        7.5. Product Transmission 59
2. User Manual 61
    1. Startup Menu 62
    2. TAF Monitor 64
        2.1. TAF Monitor Window 65
        2.2. Server Status Indicators 68
        2.3. Editor and Product Selection Buttons 70
        2.4. Menu Bar 71
        2.5. Resource Editor 73
    3. TAF Editor 75
        3.1. Starting the TAF Editor 76
        3.2. Using the TAF Viewer 77
        3.3. Using the TAF Editor 78
        3.4. Using the Guidance Viewer 83
        3.5. Taf Editor Menu Bar 86
    4. TWEB Editor 89
        4.1. Starting the TWEB Editor 90
        4.2. Using the TWEB Editor Window 91
        4.3. TAF/METAR Viewer Window 92

4.4. Menu Bar 93
5. Forecast Transmission 94
6. Weather Plot 95
7. Climatology Tools 97
7.1. Climatology Startup Menu 98
7.2. METAR Display 99
7.3. Climatological Statistics Tool 100
7.4. Climate Statistics Setup Editor 104
8. New Climatology Tools 105
8.1. Wind Rose 106
8.2. Ceiling and Visibility Distribution 109
8.3. Ceiling and Visibility Trend 112
A. Editor Key Bindings 115
B. Monitoring Rules 117
C. Climate Quality Control 123
D. Low Level Wind Shear Monitoring 127
E. Ceiling and Visibility Grid Monitoring 129
Bibliography 131

List of Tables

A.1. Key Bindings 115
B.1. METAR Monitoring Rules 117
B.2. GFE Grid Monitoring Rules 119
B.3. Lightning monitoring rules 120
B.4. Lightning probability forecast monitoring rules 120
B.5. CCFP monitoring rules 120
B.6. Low Level Wind Shear Monitoring Rules 120

Introduction

The Aviation Forecast Preparation System (AvnFPS) is a powerful AWIPS application program that helps forecasters monitor weather conditions and prepare their aviation forecasts. The AvnFPS monitoring capability gives forecasters quick and continuous feedback on TAFs as well as associated observations. This monitoring capability uses a site-configurable, color-coded scheme. AvnFPS includes specialized editors to aid the production of TAFs and TWEBs. These editors can display TAFs, guidance products, and current observations as both text and graphics. AvnFPS includes tools that help forecasters assess the quality of each forecast before it is issued. These quality control tools can assess the syntax of the forecast as well as compare its meteorological content with a database of historical observational data.

Version 3 of AvnFPS introduces a number of new guidance sources and monitoring capabilities. It also introduces conditional climatology as a quality control tool and a forecast tool. AvnFPS3.2 can simplify aviation forecast verification procedures for a WFO by transmitting forecaster ID numbers to the central NWS verification databases.

Chapter 1. System Administration Manual

This manual is intended for those who will install, configure and maintain AvnFPS.

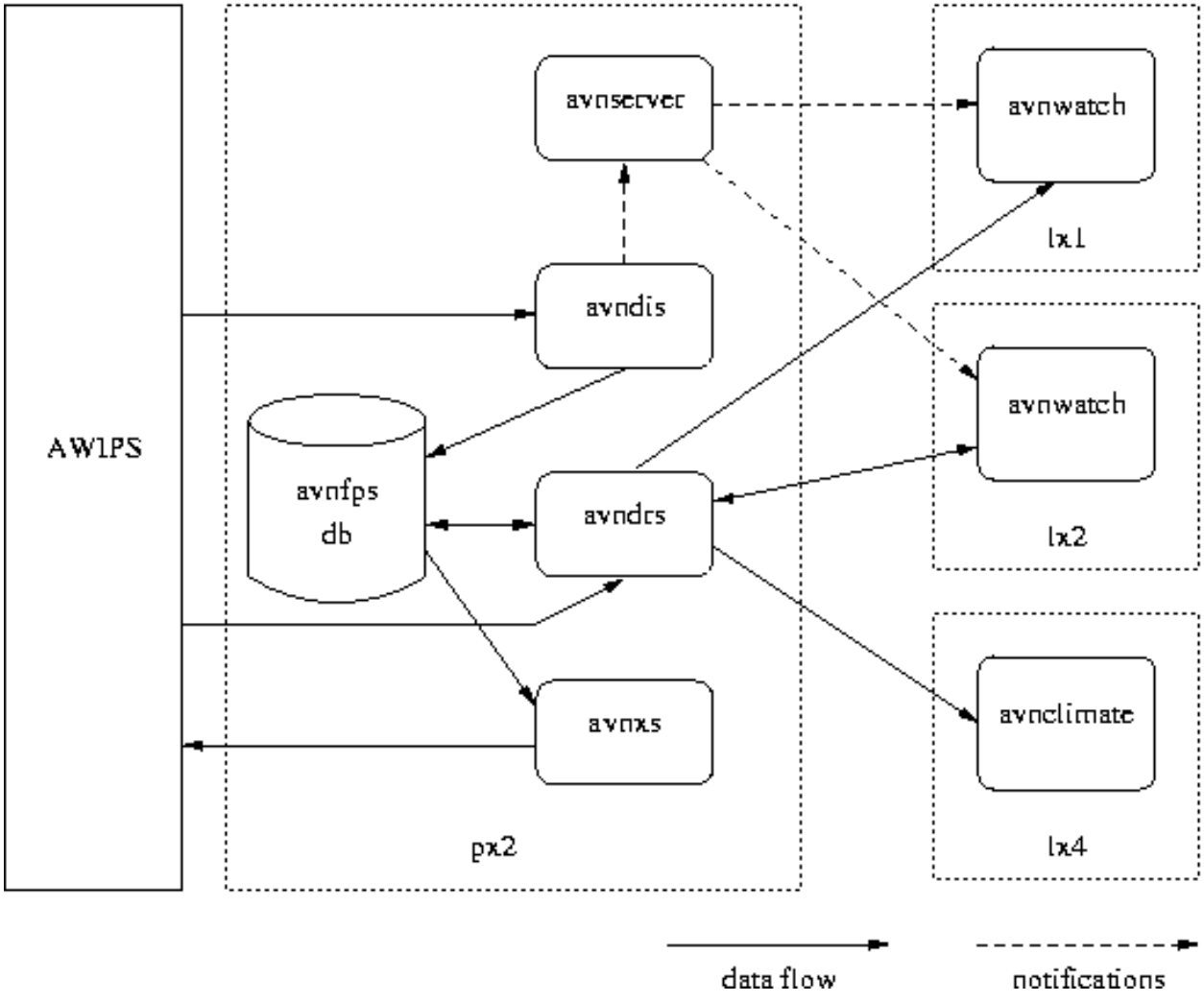
1. System Overview

This section describes technical details of AvnFPS.

AvnFPS is implemented with a series of server and client processes that communicate via TCP sockets. The servers run on hosts where the data are readily available. Clients run on workstations which do not need direct access to the data. The client may be located on a non-AWIPS system and does not have to be within the same network segment. Data request servers (**avndrs**) are responsible for delivering data to clients. Data ingest servers (**avndis**) monitor data received by AWIPS and store them in a modified format in the AvnFPS database. A notification system (**avnserver**) is responsible for alerting the clients when new data are received. The AvnFPS database can be distributed among several hosts.

AvnFPS is written mostly in the Python programming language [Python]. Remote communication features are implemented with the help of Python Remote Objects (PYRO). PYRO [Pyro] is an advanced and powerful Distributed Object Technology system, written entirely in Python, which resembles Java's Remote Method Invocation (RMI). TAF and METAR decoders utilize Toy Parser Generator, [TPG], also a Python module. Graphical interfaces are written in Tkinter. The AvnFPS distribution comes with its own version of the above tools, and is independent of AWIPS COTS.

Network Attached Storage, introduced with AWIPS OB6, eliminates the need for a distributed server configuration, as all the data are visible from a single host. The standard configuration is shown below.



Data flow diagram: AWIPS OB7

Note

The above diagram shows servers running on `px2`, however, any Red Hat™ host on a AWIPS network would do. The standard installation puts the servers on `px2`.

1.1. Name and Event Server

The **avnserver** process stores the names and locations of all objects used by AvnFPS servers, and it provides these names and locations to all other AvnFPS processes. Additionally, **avnserver** other task is to provide a notification mechanism between servers and clients. Thus, **avnserver** combines PYRO's name and event servers. One and only one instance of avnserver should run within a WFO cluster. A client first connects to **avnserver** to get information about the locations of data request servers and to subscribe for notifications. Client processes do not have to run on hosts at the same WFO as the data servers although this feature is generally not used. An access list of valid hosts, maintained in a configuration file, is used to provide host based access control. **avnserver** must be running before any other AvnFPS servers or clients can be successfully started.

The application **avninit** is used to launch AvnFPS servers and in a certain order. Users typically do not have to be concerned about the servers. It is **avninit's** job to ensure that all servers are running and restart them if they fail. However, command argument details for each server are provided here for their usefulness in troubleshooting, should it be necessary to start them by hand.

The **avnserver** program accepts the following command line arguments:
`avnserver` [-d] -n *`host`*
where *`host`* is the host name where the process will run. Obviously, this value could be retrieved from the operating system. This argument, however, is needed to properly handle failover situations. The optional flag -d tells the program to not detach itself from the terminal session, in other words, to run in the foreground.

1.2. Data Ingest Server

An instance of **avndis** monitors AWIPS data directories using **gamin** [Gamin]. **avndis** is a threaded application: for each data source, there is a dedicated thread that processes the data. Which server instance processes which data is determined by the server configuration file `etc/server.cfg` (page11).**avndis** receives a notification from the **gamin** server when a file in a monitored directory is created or modified. The server, or parent process, then passes the file name to appropriate thread. The thread processes the file's content, writes data to AvnFPS database and returns information back to the parent process. The parent process then sends notification to **avnserver**. A client that subscribed to the server will then receive this notification. Currently implemented threads are:

text	Processes text products: TAFs, METARs, TWEBs and CCFPs.
ltg	Processes lightning data.
rltg	Processes regression-based lightning forecasts data.
llws	Calculates Low Level Wind Shear values. The data sources are radar data (VWP) and METAR observations.
file	Monitors files written by an external program in a format that can be processed directly by AvnFPS clients. Currently used to process IFPS grids.
guid	Monitors directories where guidance sources arrive over the AWIPS WAN. Typically, BUFR MOS and raw model output from NCEP.

Every 30 seconds **avndis** sends an ALIVE message that is used by clients to monitor the state of AvnFPS servers and to alert users if something goes wrong.

The program accepts the following arguments:
`avndis` [-d] -n *`host`*
where *`host`* is the host name where the process will run. The optional flag -d tells the program not to detach itself from the terminal session (i.e., to run in the foreground). See Section 3.1: “Starting AvnFPS Servers” (page30) for an example of **ps** listing.

1.3. Data Request Server

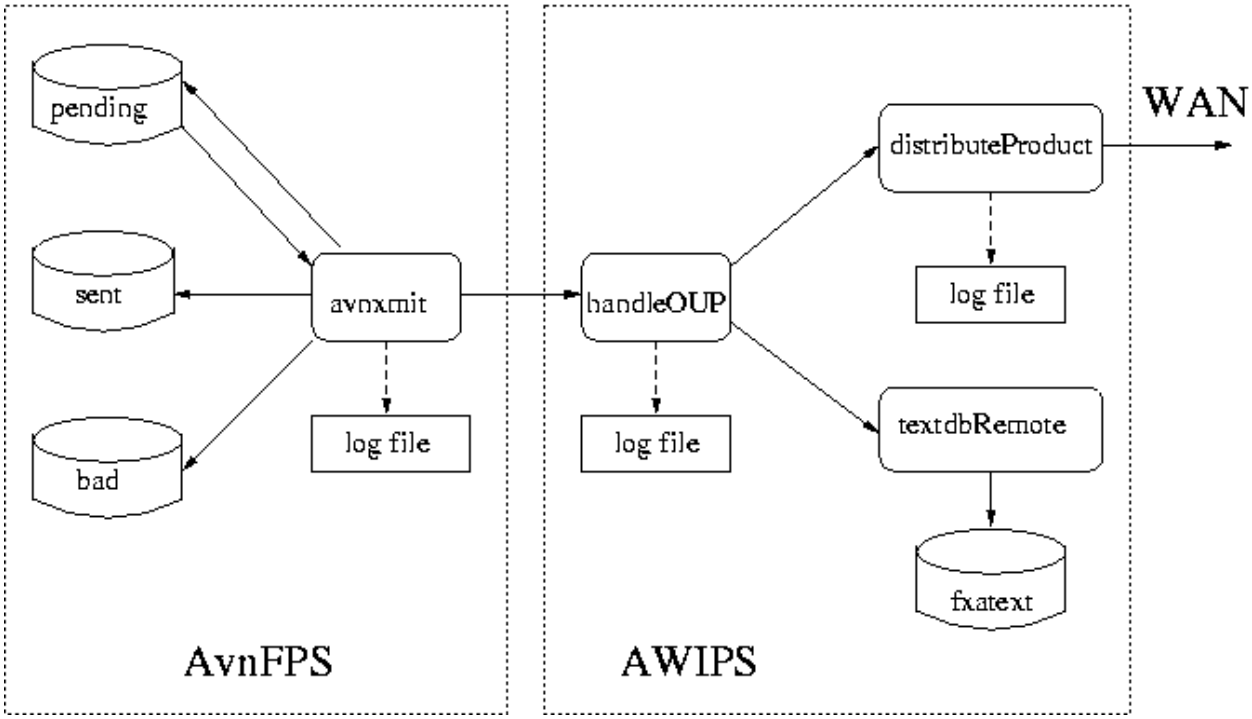
avndrs is mainly responsible for providing data to AvnFPS clients. The data source is the AvnFPS database, however, it can also directly access AWIPS files such as NetCDF data in the `/data/fxa` tree. The other function of **avndrs** is to write forecasts prepared by the clients to a queue where they are then processed by the transmission server **avnxs**. Each instance of **avndrs** is capable of processing all data sources.

Every 30 seconds **avndrs** sends an ALIVE message that is used by clients to monitor state of AvnFPS servers and alert users when something goes wrong.

The program accepts the following arguments:
`avndrs` [-d] -n *`host`*
where *`host`* is the host name where the process is running. The optional flag -d tells the program not to detach itself from the terminal session (i.e., to run in the foreground). See Section 3.1: “Starting AvnFPS Servers” (page30) for an example of **ps** listing.

1.4. Transmission Server

The transmission server, **avnxs**, provides a bridge between AvnFPS and the AWIPS transmission system. **avnxs** supports delayed transmissions, i. e., issuing forecasts that have been prepared by the forecaster before the transmission window opens.



Data flow diagram: Transmission Server

Forecasts prepared by the AvnFPS forecast editor are written to the directory `xmit/pending` as text files with a particular naming convention. The filename is then used to determine the disposition of the product. Files in the `xmit/pending` directory are named as follows:

`fid-ccccnnnxxx-wmoid-wfoid-ymmddhhmm-typ-ttttttttt`
where

fid:
: Forecaster ID Number

ccccnnnxxx:
: AWIPS id, argument to **handleOUP.pl**

wmoid:
: WMO Header

wfoid:
: WMO ID if the WFO issuing the product

yymmddhhmm:
: Full Timestamp, as year (modulo 100), month (Jan=1), day of the month, UTC hour and minute

typ:
: Forecast Type and Version, one of the following

___	Routine Forecast
RR*X*	Delayed Forecast
AA*X*	Amendment
CC*X*	Correction

where *X* is the version: A through Z.

ttttttttt:
Earliest transmission time, in Unix seconds. For routine forecasts this is the start of the transmission window or an arbitrary time set by the forecaster. For other forecasts this is the time the forecaster presses the **Send** button.

avnxs monitors files in the `xmit/pending` directory to the current system clock. The requested transmission time, T_{xmit}, and the creation time, T_{post}, are retrieved from the filename and from the OS, respectively. If the current time T_{cur} is within range: $T_{xmit} < T_{cur} < T_{post} + N$ hours, **avnxs** will call an AWIPS program (currently **handleOUP.pl**) to send the forecast out to the WAN. *N*, the number of hours before a pending file is considered 'old', is now a configurable item in `etc/xmit.cfg` (page17) file. If the forecast file is too old, it will be moved to`xmit/bad` directory. The value of *N* (usually 3 hours) is set in transmission server configuration file. Files that do not match transmission file format are discarded. **avnxs** checks for the return code from **handleOUP.pl**. The status is written to a file `xmit/`*DoW*, where *DoW* is a 3-letter abbreviation for the Day Of the Week. Here is a sample entry from `xmit/`*DoW*:
SUCCESS002-KPBZTWB072-FRUS41-KPBZ-0410270100-___-1098840000
The first word is either SUCCESS or FAIL*nnn*, and *nnn* is the error code returned by **handleOUP.pl**. If **handleOUP.pl** fails to transmit the forecast, the file will be moved to `xmit/bad`. The transmission status will be announced via **avnserver** to all client GUIs, see Section 5, “Forecast Transmission” [94].

The program accepts the following arguments:
`avnxs` [-d] -n *host*
where *host* is the host name where the process is running. The optional flag -d tells the program not to detach itself from the terminal session (i.e., to run in the foreground). See Section 3.1: “Starting AvnFPS Servers” (page30) for an example of the **ps** listing.

The transmission server also collects information on forecasts sent by the office for verification purposes. A special product is send periodically to the Verification Branch at NWS Headquarters that contains information that identifies which forecaster is responsible for which forecast. A configuration file `etc/xmit.cfg` (page17) specifies WMO and AWIPS headers and the frequency of transmission.

1.5. Client Programs

AvnWatch is the main client application. It provides forecast preparation and monitoring functionality. **Avnclimate** is a set of GUIs accessing archived climatological data. Current functionality provides display of historical data and statistics that can help by producing a short-term forecast based on current weather conditions. These applications are described in much greater detail in Chapter 2.

2. Configuring AvnFPS

This section describes directory layout, configuration files used by AvnFPS and steps needed to download climatological data. Generally, this section will only be used as an aid in troubleshooting problems with the configuration of AvnFPS. While it is possible to configure AvnFPS by editing these files directly, most people will find it much easier to use the Graphical User Interfaces (GUI) described in Section 4, “Avnsetup” [].

After successful installation of AvnFPS 3.2, the top level directory `/awips/adapt/avnfps` should be as follows:

```
lx2-wfo: /bin/ls -la
drwxrwxr-x   9 fxa     fxalpha 4096 Feb 21 15:04 3.1
drwxrwxr-x   9 fxa     fxalpha 4096 Mar  1 13:03 3.2
lrwxrwxrwx   1 root    root       7 Mar  1 13:33 bin -> 3.2/bin
lrwxrwxrwx   1 root    root       8 Mar  1 13:33 data -> 3.2/data
lrwxrwxrwx   1 root    root       7 Mar  1 13:33 etc -> 3.2/etc
drwxrwxr-x   6 fxa     fxalpha 4096 Mar  1 19:27 freetype-2.1.10
drwxrwxr-x   5 fxa     fxalpha 4096 Mar  1 16:10 gamin-0.1.7
drwxrwxr-x   6 fxa     fxalpha 4096 Mar  1 20:27 hdf5-1.6.4
lrwxrwxrwx   1 root    root       8 Mar  1 13:33 logs -> 3.2/logs
drwxrwxr-x   7 fxa     fxalpha 4096 Jul  1  2005 Python-2.4.1
drwxrwxr-x   6 fxa     fxalpha 4096 Jul  1  2005 tcltk8.4.5
```

The installation process does attempt to remove versions of the software older than 3.1. It is up to you to remove any remaining cruft that might be in this directory.

All configuration files are located in the directory `/awips/adapt/avnfps/3.2/etc` and its subdirectories. Most of the files (those with an extension `.cfg`) are in Windows™ ini format:

```
[section]
          keyword=argument(s)
```

Lines starting with # are comments.

Files containing specifications for graphical user interfaces (GUI) follow a X11 standard:

```
*pattern: value
```

where *`pattern`* is a hierarchy of one or more widget classes/names, separated by `*` or a `.`. Comment lines start with `!`. See [welch] [131], Chapter 25. These files are located in the subdirectory `app-resources`.

The file `etc/forecasters` contains forecaster IDs and names in plain ASCII text format.

2.1. Files in `etc`

`etc/forecasters`

Contains list of forecaster ids and names/initials. The numbers must be in the range 1 through 999. Lines starting with # are comments. Blank lines are allowed in the file. The format is the same as in previous releases. An example:

```
# etc/forecasters
# List of forecasters
# Format:
# number name
1 Amanda
2 Bailing
3 Belinda
```

`etc/logging.cfg`

This file configures the logging behavior of all AvnFPS programs. You should not need to modify it. Not changed since AvnFPS 3.0.

```
# logging.cfg
# configuration file for Python logging utility

[loggers]
keys=root

[handlers]
keys=hdl

[formatters]
keys=fmt

[logger_root]

level=NOTSET
handlers=hdl

[handler_hdl]

class=WeeklyFileHandler
args=('%s/logs' % os.environ['TOP_DIR'], os.path.basename(sys.argv[0]).split('.')[0])
formatter=fmt

[formatter_fmt]

format=%(asctime)s %(levelname)-5s [%(process)5d:%(thread)6d] %(module)s: %(message)s
```

❶ Log level is set in the section [logger_root]. Available values in decreasing order of information output are: DEBUG or NOTSET (these two are equivalent), INFO, WARNING, ERROR, CRITICAL.

❷ [handler_hdl] specifies output directory and file name. The class `WeeklyFileHandler` appends an abbreviated day of the week to file name.

❸ [formatter_fmt] specifies format of a line written to the log file. Refer to Python Documentation [http://docs.python.org/lib/module-logging.html] for details.

`etc/server.cfg`

Modified in release 3.2. Specifies values used by the data ingest servers.

```
# etc/server.cfg
# server configuration file for OB7
# last update: 03/01/06

# Name servers
# Either local or ip address. Local means that broadcast is used to locate
# the server, which means the server must be on the same subnet
# The list of tafs is displayed in the startup (menu) GUI
```

```
# Data ingest servers
[dis]                    ❶

tags=text2,ltg,rltg,file,llws,guid

#[dis_text]              ❷

#name=text
#source=/data/fxa/point/avnfps/raw
#module=TextThread
#nhours=24

[dis_text]               ❸

name=text
source=/awips/adapt/avnfps/data/text
module=TriggerThread
nhours=24

[dis_ltg]                ❹

name=ltg
source=/data/fxa/point/binLightning/netcdf
module=LtgThread
# distance in miles, age in minutes
distance=20
age=15

[dis_rltg]               ❺

name=rltg
source=/data/fxa/img/SBN/netCDF/LATLON/3hr/LTG
module=RadLtgThread

[dis_llws]               ❻

name=llws
radar=/data/fxa/radar
profiler=/data/fxa/point/profiler/netcdf
module=LLWSThread

[dis_file]               ❼

tag=px1
name=file
source=/awips/adapt/avnfps/data/grids,
module=FileThread

[dis_guid]               ❽

name=guid
ngmmos=/data/fxa/point/mos/NGM/netcdf
avnmos=/data/fxa/point/mos/AVN/netcdf
gfsmos=/data/fxa/point/mos/GFS/netcdf
etamos=/data/fxa/point/mos/ETA/netcdf
eta=/data/fxa/point/model/ETA/netcdf
module=GuidanceThread
nhours=24

# access control
# hosts allowed to access data
[valid]                  ❾

hosts=XXX.XXX.58.*
```

❶ Section [dis] specifies which ingest threads should run. For each item *tag* on the list there is a corresponding [dis_*tag*] section which defines the thread. Each section comprises several items listed below. There are two ways of accessing text products. TextThread reads data files written by **acqserver**. However, the AWIPS ingest system has to be changed from baseline configuration in order to use this thread. Thus, it is commented out and TriggerThread is used in AWIPS instead.

- name is one of predefined data sources which is specific to each module. Do not modify this value.
- source is the directory where AWIPS data files reside. This directory is monitored by **gamin** server daemon, gam_server. In this case, this is the directory where **avntrigger.sh** writes files extracted from text database. A special case is LLWSThread which relies on radar and profiler data. The relevant directories are specified by keywords radar and profiler.
- module specifies Python source code file. Do *not* change.

- nhours is a module specific parameter, in this case it is the number of hours of data that should be preserved in AvnFPS database.

❷ Section [dis_text] defines an experimental setup that bypasses AWIPS decoders and databases. TAFs, TWEBs and METARs are retrieved from files dumped by **acqserver**into a directory. This configuration is not recommended for novices.

❸ Item [dis_text] defines thread processing reports retrieved from Postgres fxatext by **avntrigger.sh** via the trigger mechanism. This is delivered as the default technique to get text products.

❹ ltg tag controls the processing of lightning observations. The tag distance defines an area of interest. Only lightning strikes within a square centered at the TAF site of size twice the provided value (in miles) will be extracted from AWIPS data files.

❺ [dis_rltg] tag is for the radar-based 3 hour cloud-to-ground lightning probability forecast which is currently available in the lower 48 states. OCONUS sites may safely remove this tag from the list in the [dis]section.

❻ [dis_llws] controls Low Level Wind Shear setup. The directories are the base radar file location (the site id and product type VWP will be appended by the thread). The profiler tag can be a comma-delimited list of directories. This allows LDAD directories as well as AWIPS ones to be listed.

❼ The [dis_file] describes a thread that reads the content of files in the directories listed on the source line. The base directory of each file is passed to the client, which can thus determine the data type. Currently this thread reads files produced by a GFE text formatter.

❽ The [dis_guid], specifies guidance data: MOS and high-resolution point NAM-WRF model data. You may comment out some lines if a particular data is not available in your area. For instance, NGM MOS isn't available in Alaska and Pacific Regions, so ngmmos tag can be removed.

❾ The access control section [valid] restricts access to AvnFPS server to hosts with IP addresses matching those on the hosts line. A shell type match characters can be used: * matches any string, ? matches single character

etc/px2finit.cfg

Specifies which servers should be started by **avnint** and their startup order and should not be changed.

```
# px2finit.cfg

[avnserver]     ❶

name=avnserver
order=0
wait=5

[avndrs]
name=avndrs
order=1
wait=3

[avndis]
name=avndis
order=2
wait=1

[avnxs]
name=avnxs
order=3
```

❶ For each server there is a separate section. The tag must be unique. A section consists of 2-4 items:

- name specifies server name
- order determines server startup sequence. Servers with lower order start first.
- wait is a wait time in seconds after a particular server is started (forked), before starting the next server on the list.

etc/gui.cfg

This file has been changed in 3.2. Specifies values used by graphical user interfaces (GUI) shared by all users.

```
# etc/gui.cfg
# specifies available monitor options for avnwatch

# Name servers
# Either local or ip address. Local means that broadcast is used to locate
# the server, which means the server must be on the same subnet
# The list of tafs is displayed in the startup (menu) GUI
[ns]
tags=local,

# colors in main GUI: 7 items
[colors]
tags=green3,grey,pale green,yellow,orange,red,purple

# editor tags
[editor_tags]
fatal=red
error=orange
warning=forest green

# miscellaneous features
[features]
tafeditor=1
twbeditor=1

# viewers: taf and metar must be present, taf must be first on the list
[viewers]
tags=taf,metar,gfsmos,etamos,ngmmos,etabuf,grids

[viewer_taf]
module=TafViewer

[viewer_metar]
module=MetarViewer
label=Metars

[viewer_gfsmos]
module=MosViewer
label=GFS-MOS
model=gfsmos

[viewer_gfslamp]
module=MosViewer
label=GFSLAMP
model=gfslamp

[viewer_ngmmos]
module=MosViewer
label=NGM-MOS
model=ngmmos

[viewer_etamos]
module=MosViewer
label=ETA-MOS
model=etamos

[viewer_etabuf]
module=EtaViewer
label=NAM-WRF-profile

[viewer_grids]
module=GridViewer
label=Grids

[viewer_avnmos]
module=MosViewer
label=AVN-MOS
model=avnmos

[menus]
number=7

[menu_0]
items=metar

[menu_1]
items=persistence_1hr,persistence_2hr,persistence_3hr

[menu_2]
items=ltg

[menu_3]
items=rltg
```

```
[menu_4]
items=ccfp

[menu_5]
items=grids

[menu_6]
items=llws

[monitor_metar]

menu=METAR
module=MetarMonitor
items=tempo,vsby,wind,wx,sky
labels=tpo,vis,wnd,wx,cig

[monitor_persistence_1hr]
menu=persistence 1hr
module=PersistMonitor
nhours=1

items=vsby,wind,wx,sky
labels=vis,wnd,wx,cig

[monitor_persistence_2hr]
menu=persistence 2hr
module=PersistMonitor
nhours=2
items=vsby,wind,wx,sky
labels=vis,wnd,wx,cig

[monitor_persistence_3hr]
menu=persistence 3hr
module=PersistMonitor
nhours=3
items=vsby,wind,wx,sky
labels=vis,wnd,wx,cig

[monitor_ltg]
menu=ltg
module=LtgMonitor
items=wx
labels=ts

[monitor_rltg]
menu=rltg
module=RadLtgMonitor
items=wx
labels=ts

[monitor_ccfp]
menu=ccfp
module=CCFPMonitor
items=wx
labels=ts

[monitor_grids]
menu=grid
module=GridMonitor
from=2

to=6
items=vsby,wind,wx,sky
labels=vis,wnd,wx,sky

[monitor_llws]
menu=llws
module=LLWSMonitor
items=wind
labels=ws
```

❶ Section [colors] lists colors used to indicate discrepancies between forecasts and observations/guidance, in increasing order of severity.

❷ Section [viewers] lists data viewers available in the TAF Editor window. For each tag listed, there must be a [viewer_tag] section.

❸ Each viewer requires certain parameters. module specifies a Python source code file, label defines text displayed in the window's tab. Some modules require additional values. You should not modify any of those values, with the exception of the labels.

❹ Section [menus] defines data areas in the main monitoring GUI. For each area there must be a section [menu_#] listing items that will be monitored. Count starts from 0.

❺ Items is a list of comma separated tags, each of the tags must have a corresponding [monitor_item] section. If the list consists of one item, it *must* be terminated by a comma. If 2 or more items are on the list, the user can

select what is monitored.

❻ Each monitor requires certain parameters. module specifies a Python source code file, menu defines text displayed when the popup menu is invoked. items is a list of monitored weather elements, labels determine displayed text. To keep the GUI nicely aligned, keep label names 2 or 3 characters long.

❼ nhours is the number of hours ahead assumed the weather does not change

❽ from, to define interval of time the grids are monitored against the forecast

etc/wxplot.cfg

This is a configuration file for the weather plot GUI.

```
# wxplot.cfg

[print]                    ❶
cmd=convert - tmp/%s.jpg

[hours]                    ❷

back=24
forward=24

[vsby]                     ❸

bot=0.125
top=10.0

[cig]
bot=100
top=8000

[viewers]                  ❹

tags=taf,metar,gfsmos,etamos,ngmmos,eta

[selected]                 ❺

tags=taf,metar

[viewer_taf]               ❻

module=TafPlot
label=TAF
color=blue

[viewer_metar]
module=MetarPlot
label=METARs
color=red

[viewer_gfsmos]
module=MosPlot
label=GFS-MOS
color=forest green
model=avn

[viewer_etamos]
module=MosPlot
label=ETA-MOS
color=green
model=gfs

[viewer_ngmmos]
module=MosPlot
label=NGM-MOS
color=dark green
model=ngm

[viewer_eta]
module=EtaPlot
label=NAM-WRF
color=brown
```

❶ Command to be executed when the **Print** is pressed. A window dump file (xwd format) is passed via standard input stream.

❷ Number of hours of data to display, counted from the current hour.

❸ Ceiling and visibility plot limits.

❹ This section lists data viewers available in the Weather Plot window. For each tag listed, there must be a [viewer_tag] section.

❻ Data viewer toggles selected by default.

❻ Each viewer requires certain parameters. 'module' tag specifies a Python source code file, 'label' tag defines text displayed as a toggle button name. Some modules require additional values. You should not modify any of those values, with the exception of the labels.

etc/xmit.cfg

This configuration file is used by **avnxs** which sends forecast and verification products out for dissemination.

```
# etc/xmit.cfg

[verification]          ❶

wmo = NXUS98 KPIT
awips = KPBZVFTPBZ
fcstid = 0
period = 6

[transmission]          ❷

# how often checks for new files (seconds)
frequency = 15
# reject older than (hours)
old = 3
```

❶ This section defines parameters used by the program to transmit the NXUS98 verification product to NWS headquarters. wmo and awips are the WMO and AWIPS headers. These headers must exist in the file `/awips/fxa/data/afos2awips.txt`. The transmission file name contains (reserved) forecaster id 0. This should not be changed. The verification product is transmitted every 6 hours.

❷ You may want to adjust the old value. It defines the cutoff time for early transmission.

etc/triggerTemplate

Made by **avnsetup**. This file is processed during trigger localization on the data server. **avnsetup** also creates a file `etc/triggerTemplate.good` while updating Postgres triggers directly. The latter is *not* used by the localization script.

2.2. TAF configuration files

The TAF configuration files reside in directory `etc/tafs`. For each TAF site CCCC there is a subdirectory `etc/tafs/CCCC` containing template files, one per each forecast period and site info data `etc/tafs/CCCC/info.cfg`.

Site Info File

This file is normally created by the Taf Site Editor GUI, normally invoked from **avnsetup**.

```
# etc/tafs/KPIT/info.cfg

[headers]
wmo = FTUS41 KLWX
afos = WBCTAFIAD

[thresholds]                    ❶

cig = 200,600,1000,2000,3100
vsby = 0.5,1.0,2.0,3.0,6.0
radar_cutoff = 600,0
profiler_cutoff =

[sites]                         ❷

metar = KIAD
eta = KIAD
avnmos = KIAD
ngmmos = KIAD
gfsmos = KIAD
gfslamp = KIAD
radars = KLWX,KDOX
profilers =

[geography]                     ❸

lat = 38.96
runway = 60,90,
lon = -77.45
elev = 98
```

❶ Section [thresholds] defines ceiling and visibility categories used by the monitoring system. A new feature of AvnFPS 3.2 is the cutoff thresholds for the VWP sources. Cutoff values correspond to heights *in meters* below which data will be ignored. As in the example above, for the KLWX radar, any VWP data below 600 meters will be ignored. However, all VWP data from the surface up to 200O feet from the KDOX radar will be used. This feature is useful if your VWP source(s) is (are) routinely contaminated by non-meterological phenomenon near the site.

❷ Section [sites] defines IDs used by other data sources. Some TAF sites may have more than one associated observation. In that case, separate the ids by a comma.

❸ Section [geography] provides location of the airport. elev is in *meters*, runway directions in degrees *true north* are used to calculate cross winds.

Important

To correctly alert forecasters on crosswinds, remember that runways are identified based on magnetic north at the time they were constructed, not true north. For much of the CONUS, SJU, HFO and GUM, with rounding to the nearest 10 degrees for identifying runways, and magnetic declinations less than 20 degrees this correction may not make much of a difference. However, for airports in the Pacific Northwest, Alaska and the Northeast, the magnetic declination is significant. See the National Geomagnetism Program [http://www.geomag.usgs.gov] web site for historical magnetic declinations for any locale.

Monitoring Rules

The default configuration files for all monitors are located in the directory `etc/tafs/XXXX`. Currently these are: `grids.cfg`, `ltg.cfg`, `mtrs.cfg`, `rltg.cfg`, `ccfp.cfg`. All can be edited with the "Montoring Rules" GUI

via **avnsetup**. Here is an example of rule definition file for monitoring of METARs:

```
# etc/tafs/XXXX/mtrs.cfg

[rule_11]          (1)

severity = 5
msg = Visibility difference of 3 or more categories
unique = 1
type = vsby
method = VsbyCatDelta
ncat = 3
.......
[rules]            (2)

active = 0,1,2,3,4,5,6,7,8,9,10,11
.......
[rule_1]
severity = 3
msg = Freezing precipitation in TAF and not in obs
unique = 0
type = wx
method = WxTafDelta
wx = FZRA,FZDZ,PL
.......
```

(1) Each [rule_*N*] provides arguments to the monitoring function, specified by the method keyword. The following tags must exist for each rule:

1. severity: a number in the range 1 through 5 corresponding to colors defined in `gui.cfg`, see (1) [14].
2. msg: message displayed by the monitoring GUI.
3. type: one of wind, vsby, wx, sky
4. unique: 0 or 1.

Each function may accept additional arguments, such as ncat. In this case ncat is the number of categories the visibilities in a TAF/METAR pair must differ for the rule to activate. Thus you may specify the same method with different arguments, ncat=2 in our case.

(2) Section [rules] defines the list of rules to monitor. For each number there must be a corresponding [rule_*N*] section.

If you want to have rules dependent on the TAF site, copy the relevant file to the TAF site directory and modify it there. The monitoring program first tries to find the files in the site directory, if it can't find it there, uses the one from `etc/tafs/XXXX`.

Important

There are no default hardcoded rules, the files in `etc/tafs/XXXX` must exist.

Impact QC Definition File

Impact QC is a new feature in AvnFPS3.2. The forecaster can be alerted when a prepared TAF has a significant impact on airport operations. This file is optional. Currently this file cannot be modified with **avnsetup**, you must use a text editor instead. Shown below is a configuration file with additional rules added that are specific to Pittsburgh International Airport, `etc/tafs/KPIT/impact.cfg`:

```
# impact.cfg

[conditions]       (1)

items=cond_1,cond_2,cond_3,cond_4,cond_5

[cond_1]           (2)

tag=FC
level=1
```

```
text=Impact: Fuel-Alternate vsby<3 or cig<2000
expr=vsby<2.95 or cig<2000

[cond_2]
tag=FC
level=2
text=Impact: LIFR conditions vsby<1 or cig<500
expr=vsby<0.95 or cig<500

[cond_3]
tag=ws
level=1
text=Wind shift vs previous group resulting in >6kt change along PIT runway 28R/10L,may affect runway config
expr=wind[0].shift and wind[0].runway>6

[cond_4]
tag=xw
level=1
text=Cross wind component >14KT on PIT runway 28R/10L
expr=wind[0].cross>14

[cond_5]
tag=xw
level=2
text=Cross wind component >24KT on PIT runway 28R/10L
expr=wind[0].cross>24
```

❶ Section [conditions] defines the list of conditions to check. For each tag [cond_*N*] there must be a corresponding section.

❷ Each [cond_*N*] provides arguments to the impact check function. The check is done for each TAF period. The following tags must exist for each rule:

1. expr: a Python expression to be evaluated. This expression can contain following variables: vsby, cig and wind. vsby is in miles, ceiling in feet. wind is an array that has length equal to the number of runways. Each element of this array contains 3 values: runway, cross and shift. The wind vector is represented in a Cartesian coordinates along and perpendicular to the runway. The absolute values of these components are wind[*N*].runway and wind[*N*].cross, where *N* is runway number, counted from 0, as listed in the site configuration file, see section: : “Site Info File” (page 18). wind[*N*].shift is a Boolean value, indicating that the runway component changed sign from one TAF period to another.
2. text: message to be displayed when the expression evaluates to True.
3. level: highlight background color, 1 - green, 2 - orange.
4. tag: used to group messages, so only the message with the highest level is displayed. In this example, if visibility is 1/2SM, only the LIFR message is shown in a popup window of the TAF Editor.

Climate QC Definition File

The file `etc/tafs/XXXX/climqc.cfg` is used to define category thresholds used while creating climate QC count files. Currently this file cannot be modified with **avnsetup**, you must use a text editor. An example:

```
#etc/tafs/XXXX/climqc.cfg

[args]          ❶

alpha=0.3
showdetails=0

[thresholds]          ❷

vsby=0.27,1.1,2.9,6.1
cig=205,605,1005,3105,40000
ff=4.1,10.1,20.1
dd=22.5,67.5,112.5,157.5,202.5,247.5,292.5,337.5
```

❶ The parameter alpha is used to determine likelihood of occurence of a combination of weather elements in the forecast based on conditional probabilities. Alpha should be a positve number, less than one. The showdetails flag (0=no;1=yes) turns detailed statistics display, generally disliked by forecastes.

❷ The thresholds determine category ranges. Each of arguments is a list of values $v_1 < v_2 < ... < v_n$, with implicit $v_1 = 0$ and $v_{n+1} = \#$. Category k consists of all values v such that $v_k \# v < v_{k+1}$.

The value 40 000 ft on the ceiling list serves to distinguish between limited and unlimited ceiling, which is coded as 99 998 ft.

If you want to have thresholds dependent on the TAF site, copy the file to the TAF site directory and modify it there. The program **avnqcstats** which creates the count files first tries to find the file in the site directory, if it can't find it there, uses the one from `etc/tafs/XXXX`.

TAF Product Definition Files

Located in `etc/tafs`. Normally created by the Taf Product GUI invoked from **avnsetup**. An example file:

```
# etc/tafs/PBZ_TAFS.cfg

# list of TAF sites
[sites]
workpil = PITWRKTAF
idents=KPIT,KAGC,KBVI,KLBE,KHLG,KMGW,KZZV,KFKL,KDUJ
```

Default Product File

The optional `DEFAULT` file contains name of the configuration file, *sans* the `.cfg` extension. that should be loaded on startup of the monitoring GUI. If not present, the first one in alphabetical order is used.

TAF Formatter Configuration Files

The TAF Formatter requires three configuration files. These files are located in the directory `etc/tafs/XXXX`. With AvnFPS3.2, `flt_cat.cfg` and `grp_taf.cfg` files can now be customized for each TAF site and moved to its respective `etc/taf/CCCC` directory.

File `flt_cat.cfg` contains the threshhold values of visibility and ceiling height that are used to determine aviation flight categories. The default values are identical to those provided in Directive 10-813. The thresholds can be changed to site-specific values.

```
#etc/tafs/XXXX/flt_cat.cfg
[cig]
thresholds=200,500,1000,3000,3500,4000,5000,8000,10000,12000,15000,18000,20000
[vis]
thresholds=0.5,1,3,5 ❶
```

❶ These thresholds divide the spectrum of ceiling and visibility into categories with an implied 0 at the beginning and 'infinity' at the end of each list. Thus as defined here, ceiling category 1 corresponds to cloud heights from 0 to 200 feet, category 2 from 200 to 500 feet, etc. Similarily for visibility, with category 1 corresponding to visibilities LTE to 1/2 mile. Category 5 corresponds to visibility greater than 5 miles.

File `grp_taf.cfg` contains the configurable threshold values that are used to form concise TAFs. Detailed description for each category has been provided inside the file.

```
#
# This file contains thresholds that control the algorithm that converts
# numerical/tabular gudiance into 'guidance' TAFs.
#
# Probabilities (POT and POP06) above which precipitation and/or
# thunderstorms will be included in the prevailing group.
#
[prev]
pop=80
tstm=80
#
# Probabilities (POT and POP06) above which the tempo group is formed
# containing precipitation and/or thunderstorms.
#
[tempo]
```

```
tstm=36
pop=36
#
# Boolean (yes/no) if PROB30 groups are wanted in guidance TAFs
[prob30]
value=yes
#
# The longest time span in hours that makes this group combinable with
# a longer, adjacent (in time) group. If the duration of a group is longer
# than this value, this group stays alone in the final TAF.
#
[short_dt]
value=1
#
# The shortest time span in hours that allows its shorter duration
# neighbor to be combined with it.
#
[long_dt]
value=15
#
# Maximum category difference between the long and the short duration groups. If the
# difference in flight category between the two groups is greater than this value,
# they are not combined.
#
# Categories are treated separately for visibility and ceilings.
#
# To see the breakpoints for the ceiling and visibility categories, see
# etc/tafs/XXXX/flt_cat.cfg (flight category) configuration file.
#
# First the visibility
#
[dc_vis]
value=1
#
# Lowest visibility category to which the combining algorithm is applied
#
[low_c_vis]
value=5
#
# Highest visibility category to which the combining algorithm is applied
#
[high_c_vis]
value=5
#
# Now, the ceiling . . .
#
[dc_cig]
value=1
#
# The lowest ceiling category to which the combining is applied
#
[low_c_cig]
value=13
#
# Highest ceiling category to which the combining is applied
[high_c_cig]
value=14
#
# Earliest forecast time in hrs to which the combining algorithm can be applied
[low_p]
value=12
#
# Maximum allowable ceiling height in hundreds of feet when thunderstorms
# are forecasted
#
[cb_hi]
value=50
#
# Maximum speed which is considered to be not a calm wind.
[wind]
calmwd=5
#
# Wind speed and direction change in wind speed above which combining will not be applied.
value=10
delta=4
```

Note

Items 1 through 9 control how two *dissimilar* (in visibility and/or ceiling) groups, one 'short', the other 'long', can be further combined.

❶ The maximum duration of the smaller group that can be combined with the larger, dissimilar one. When combined, the larger group values are not adjusted by the smaller group's values.
❷ The minimum length of time to be considered the 'large' group candidate for combining with smaller ones.
❸ The largest visibility category *difference* allowed between TAF groups and still be able to combine.
❹ The lowest visibility category allowed to be considered for combining. This value is related to the values assigned to the vis thresholds in `flt_cat.cfg` file.
❺ The highest visibility category allowed to be considered for combining. This value is related to the values assigned to the vis thresholds in `flt_cat.cfg` file. Usually set to the number of visibility 'breakpoints' in vis thresholds.
❻ The largest ceiling category *difference* allowed between TAF groups and still be able to combine.
❼ The lowest ceiling category allowed to be considered for combining. This value is related to the values assigned to the cig thresholds in `flt_cat.cfg` file.
❽ The highest ceiling category allowed to be considered for combining. This value is related to the values assigned to the cig thresholds in `flt_cat.cfg` file. Usually set to the number of ceiling 'breakpoints' in cig thresholds.
❾ Number of hours when the combining algorithm will be applied to groups valid beyond the TAF's beginning valid time.
⓫ Maximum allowable difference in speed (knots) and direction (tens of degrees) between groups allowed to be combined.

Finally, `grid_probs.cfg` contains configuration data on how the probability attributes on IFPS grids are translated into numerical probability values that will be used to form TAFs. Remember that precipation is only included in the TAFs when certain probability threshold is exceeded. This file cannot be customized to individual TAF sites.

```
#tables used to convert prob symbols in grids to a pop value
[before9hr]
S=30
IS=30
WS=30

SC=50
O=50
C=50

D=70
WP=70
NM=70
L=70

[after9hr]
S=20
IS=20
WS=20

SC=30
O=30
C=30

D=50
WP=50
NM=50
L=50
```

2.3. TWEB configuration files

The TWEB configuration files reside in `etc/twbs`.

Route Configuration Files

For each TWEB route or synopsis there is a subdirectory `etc/twbs/NNN` containing template file route/synopsis info data `etc/twbs/NNN/info.cfg`. This file is normally created by the setup program **avnsetup**. An example file:

```
# etc/twbs/073/info.cfg
[headers]
wmo = FRUS41 KPBZ
afos = PITTWB073

[sites]
metars = KCLE,KYNG,KPIT,KLBE,KJST,KAOO,KMDT,KMRB,KDCA
tafs = KCLE,KYNG,KPIT,KLBE,KJST,KAOO,KMDT,KMRB,KDCA
```

Section [sites] lists TAFs and METARs that should be displayed in the TWEB Editor window.

TWEB Product Configuration Files

These files are located in `etc/twbs`. They are normally created by **avnsetup**. An example file:

```
# etc/twbs/PIT_TWEBS.cfg

[sites]
workpil = PITWRKTWB
idents = 072,073
```

Default Product File

The optional `DEFAULT` file contains name of the product that should appear selected in the product loader in the TWEB Editor window. If not present, the first product in alphabetical order is chosen.

2.4. X resources configuration files

The default and individual resource files are located in the directory `etc/app-resources`. To edit the default file `etc/app-resources/X` you can use any text editor. Individual files have naming convention `X.N`, where N is the forecaster number from `etc/forecasters`. These files are created by a graphical editor available from the AvnWatch GUI menu.

Here is the listing of the default file `etc/app-resources/X`

```
!! default X resources configuration file
!! last modified July 10, 2005
!! user editable options
!! default font
!! change 100 to 140 if you want bigger default font
*font:                          -adobe-helvetica-bold-r-normal-*-*-100-*-*-p-*-iso8859-9
!! default colors
*background:                    grey70
*foreground:                    black
!! text window font
*Text.font:                     7x14
!! text window colors
*Text.background:               #2f3f6f
*Text.foreground:               #ffffff
!! other text window specific options
*Text.width:                    74
*Text.height:                   24
*Text.cursor:                   hand1
!! forecast editor font - may want bigger
*textEditor.font:               8x16
!! forecast editor colors - may want different
*textEditor.background:         #1f2f6f
*textEditor.foreground:         white
*textEditor.width:              70
*textEditor.height:             12
!! cursor width and color
*textEditor.cursor:             arrow
*textEditor.insertWidth:        5
*insertBackground:              yellow
!! text viewer window options
*textViewer.width:              74
*textViewer.height:             12
!! forecast editor orientation: horizontal or vertical
*orientation:                   vertical
!! listbox font
*Listbox.font:                  7x14
!! entry font
*Entry.font:                    7x14
!! entry window colors
*Entry.background:              white
*MessageBar*Entry.background:   grey85
!! balloon message (popup) font
*Balloon.Label.font:            7x14
!! modifies behavior of dialogs
*transientDialogs:              1
!! confirmation on closing editor
*confirmClose:                  0
!! confirmation on sending amendments/corrections
*confirmSend:                   1
!! alert options: use colors for the first level that should result in
!! notification - pale green, yellow, orange, red, purple or none
*notifyDeiconify:               yellow
*notifyRaise:                   red
*notifyPlay:                    pale green
!! not implemented
!!*notifyTalk:                    none
*playFile:                      /awips/fxa/data/sounds/asterisk.au
!! blink on new notification
*blink:                         1
!! warning/error level to disallow send:
!! 'always', 'warning, 'error', 'fatal'
*disallowSend:                  error
!! forecast editor options
!! use template, one of: template, merge, latest
*loadOrder:                     merge
!! ask for transmission time when sending routine forecasts
*asktime:                       1
!! periodically save bulletins in a backup file
*autosave:                      1
!! update issue and valid times on QC
*updatetimes:                   1
!! print forecasts on send
*autoprint:                     0
!! Insert/Overwrite
*insert:                        1
!! Word-wrap long lines: either none or word
```

```
*wrap:                          word
!! Amd, Rtd, Cor buttons on the right
*amdbuttons:                    1
!! number of TAFs to display
*numTafs:                       1
!! number of hours of METARs to display
*numHours:                      6
!! show bulletin headers in the text window
*showHeaders:                   1
!! show decoded METARs
*showDecoded:                   1
!! show category probabilities for MOS
*showProbs:                     1
!! Format of MOS and Grids reports: raw, long or short
*showFormatted:                 short
!! flight category highlights
*highlightFlightCat:            0
*lifrColor:                     #6c1916
*ifrColor:                      #5b3c2c
*mvfrColor:                     #2c484c
*vfrColor:                      #112346
!!
!! editable by knowledgable people only
!!
!! send forecasts in a collective
*collective:                    0
!! prevents line wrap in message boxes
*wrapLength:                    0
```

Lines starting with "!!" are comments. Resource names begin with a "*" and end with ":". Anything after a ":" is a resource value. Many of the allowable values are not well documented, or cryptic. For example, valid values for cursor shape in a text window

```
*Text.cursor:                   hand1
```

are defined in file `/usr/include/X11/cursorfont.h`. You have to know that the leading XC_ in line

```
#define XC_hand1 58
```

must be stripped and 58 ignored. Valid color names are in `/usr/lib/X11/rgb.txt`, but if you don't find one you like, a hexadecimal number can be used instead, such as

```
*lifrColor:                     #6c1916
```

We suggest that you use the graphical resource editor available from in **avnsetup** before making changes to this file.

Climate applications introduced in AvnFPS 3.2 have thir own resouce files: `XWindRose`, `XCigVisTrend` and `XCVMonthly`. These are not user-specific.

2.5. Climatological Data

AvnFPS 3.2 introduces a new set of climatological GUIs requiring updated climatological files. The GUIs require fast, efficient access to voluminous amounts (often 30 years or more of data. The existing NetCDF, delivered with AvnFPS 3.1 was determined too slow and inefficient when GUI response times needed to be in seconds. HDF5 [http://hdf.ncsa.uiuc.edu/whatishdf5.html] is a file format that emphasizes storage and I/O efficiency to meet performance requirements. The AvnFPS team decided to use this format for its new climatology GUIs.

The climatological data files for all locations where the NWS prepares TAFs have been posted to a web page that is available to the World Wide Web. The Universal Resource Locator (URL) for the directory where they are posted is http://www.mdl.nws.noaa.gov/~avnfps/data/hdf5 [http://www.mdl.nws.noaa.gov/~avnfps/data/hdf5/index.html]. File names are of the form *CCCC*.hd5, where *CCCC* is the location identifier where the data were observed. The files are rather large but HDF5 already stores the data in compressed format, so there is no post-processing step after downloading it. The files are ready to be used at this point by the new climatological GUIs.

Some WFOs may find this hosting solution somewhat awkward. It seemed, however, to be the safest way to protect the AWIPS Wide Area Network (WAN) from the impacts of large data transfers.

The following procedures can be used to download the data files and make them available to AvnFPS:

Procedure 1.1. To download from an internet browser

1. On a host that can access the WWW, launch a browser application and direct it to http://www.mdl.nws.noaa.gov/~avnfps/data/hdf5 [http://www.mdl.nws.noaa.gov/~avnfps/data/hdf5/index.html]. You will see a page that lists available stations along with file modification times and sizes. The link for each station points to a binary file.

2. Find the station whose data you want to download.

3. Use the appropriate features of the browser to download the file for the station your site needs. On most browsers, this involves performing a right click on the station name and a feature such as **Save Link Target As...** or **Save Link As..**

4. Move this file to AvnFPS climate directory `/data/adapt/avnfps/climate`. Files can be moved physically via removeable media, transferred through the AWIPSPUB firewall, or transferred via LDAD.

5. Once the save process is completed you may proceed to other stations using the same procedure.

Procedure 1.2. To download using wget command on a Linux host

1. Identify the URLs for each file you will download. They are of the form http://www.mdl.nws.noaa.gov/~avnfps/data/hdf5/*CCCC*.hd5, where *CCCC* is the Station ID.

2. Identify a directory on your local system where the files can reside temporarily.

3. Use the **wget** to download each file.
   `wget` --passive-ftp -nv *dataURL* -o *output_dir/CCCC*.hd5
   where *dataURL* is the URL from the previous step, *CCCC* is the site id you and *output_dir* is the directory where you want to save the files.

4. Move these files to AvnFPS data directory `/data/adapt/avnfps/climate`. This directory is visible

from all hosts. Files can be moved physically via removeable media, transferred through the AWIPSPUB firewall, or transferred via LDAD.

3. Starting and Stopping AvnFPS

All AvnFPS programs: servers, clients and utilities are started by a shell script **avnstart.sh**. This script is responsible for setting appropriate environmental variables required by AvnFPS. The variables are actually set in the file `bin/avnenv.sh` which is sourced by **avnstart.sh**. The first argument is program name (a Python script residing in the directory `py` *without* the extension `.py`). Remaining arguments, if any, are passed to the program without modification.

- To start **avnsetup** from the command line, enter:

```
lx1-wfo: /awips/adapt/avnfps/bin/avnstart.sh avnsetup
```

- To start **avnmenu** from the command line, enter:

```
lx1-wfo: /awips/adapt/avnfps/bin/avnstart.sh avnmenu
```

3.1. Starting AvnFPS Servers

AvnFPS requires four servers. These are:

- **avnserver**: Helps client processes find the services they need (name service) and notifies clients when data are available (event service).
- **avndis**: Data Ingest Server. Copies data from various AWIPS sources into AvnFPS directory tree, reformatting as needed.
- **avndrs**: Data Request Server. Answers requests for data from client processes.
- **avnxs**: Forecast Transmission Server. Manages transmit queues and the interface to the AWIPS software that transmits products.

The order in which the servers are started is important. **avnserver** must be started first. There can be only one instance of avnserver within WFO subnet; running it on both `px1` and `px2` will lead to unexpected results. **avndis** depends on **avndrs** to process Low Level Wind Shear data, so **avndrs** needs to be started first. The simplest way to avoid these problems is to not launch these servers directly. The application **avninit** searches for active instances of all four AvnFPS servers and launches the ones that are not running. This application is modelled on the Unix™ SYSV **init** and the AWIPS **DataController**. Like **init** and **DataController**, **avninit** is a persistent process.

Note

avninit will attempt to restart a failed server 10 times within an hour. When the number of restarts exceeds 10, **avninit** will stop trying to restart the failed server.

avninit is invoked by the AWIPS startup script **px2apps** together with other AWIPS applications. If you have to start the servers afterwards, use the shell script **remoteServers.sh**. The script accepts the following command line arguments:
`remoteServers.sh` [start | stop | restart]

This script has to be run as user fxa. It uses **ssh** to start/stop **avninit** on `px2`. Therefore it can be run from any workstation. It is located in the directory `/awips/adapt/avnfps/bin`. Example usage:

```
lx2-wfo: /awips/adapt/avnfps/bin/remoteServers.sh start
```

To list avnfps processes use the **ps** command. A typical output on `px2` is:

```
px2-wfo: ps -efw | grep avnpython
fxa      11280     1 11280  0    1 Mar  01 ?        00:00:00 avnpython /awips/adapt/avnfps/3.2/py/avninit.py
px2f
fxa      11498 11280 11498  0   10 Mar  01 ?        00:00:00 avnpython /awips/adapt/avnfps/3.2/py/avnserver.py
-d -n px2f
fxa      11500 11280 11500  0    4 Mar  01 ?        00:00:00 avnpython /awips/adapt/avnfps/3.2/py/avndrs.py -d
-n px2f
fxa      11508 11280 11508  0   10 Mar  01 ?        00:00:00 avnpython /awips/adapt/avnfps/3.2/py/avndis.py -d
-n px2f
fxa      11509 11280 11509  0    1 Mar  01 ?        00:00:10 avnpython /awips/adapt/avnfps/3.2/py/avnxs.py -d
-n px2f
```

If you wish to see all the threads in hierarchical format, add -L option to **ps**:

```
px2-wfo: ps -efL | grep avnpython
fxa      11280     1 11280  0    1 Mar  01 ?        00:00:00 avnpython /awips/adapt/avnfps/3.2/py/avninit.py
px2f
fxa      11498 11280 11498  0   10 Mar  01 ?        00:00:00 avnpython /awips/adapt/avnfps/3.2/py/avnserver.py
-d -n px2f
fxa      11498 11280 11499  0   10 Mar  01 ?        00:00:00 avnpython /awips/adapt/avnfps/3.2/py/avnserver.py
-d -n px2f
fxa      11498 11280 11501  0   10 Mar  01 ?        00:00:00 avnpython /awips/adapt/avnfps/3.2/py/avnserver.py
```

```
-d -n px2f
fxa      11498 11280 11502  0   10 Mar  01 ?        00:00:00 avnpython /awips/adapt/avnfps/3.2/py/avnserver.py
-d -n px2f
fxa      11498 11280 11507  0   10 Mar  01 ?        00:00:03 avnpython /awips/adapt/avnfps/3.2/py/avnserver.py
-d -n px2f
fxa      11498 11280 11515  0   10 Mar  01 ?        00:00:04 avnpython /awips/adapt/avnfps/3.2/py/avnserver.py
-d -n px2f
fxa      11498 11280 11518  0   10 Mar  01 ?        00:00:10 avnpython /awips/adapt/avnfps/3.2/py/avnserver.py
-d -n px2f
fxa      11498 11280 11519  0   10 Mar  01 ?        00:00:00 avnpython /awips/adapt/avnfps/3.2/py/avnserver.py
-d -n px2f
fxa      11498 11280 11521  0   10 Mar  01 ?        00:00:40 avnpython /awips/adapt/avnfps/3.2/py/avnserver.py
-d -n px2f
fxa      11498 11280 11523  0   10 Mar  01 ?        00:00:18 avnpython /awips/adapt/avnfps/3.2/py/avnserver.py
-d -n px2f
fxa      11500 11280 11500  0    4 Mar  01 ?        00:00:00 avnpython /awips/adapt/avnfps/3.2/py/avndrs.py -d
-n px2f
fxa      11500 11280 11504  0    4 Mar  01 ?        00:00:00 avnpython /awips/adapt/avnfps/3.2/py/avndrs.py -d
-n px2f
fxa      11500 11280 11505  0    4 Mar  01 ?        00:00:00 avnpython /awips/adapt/avnfps/3.2/py/avndrs.py -d
-n px2f
fxa      11500 11280 11524  0    4 Mar  01 ?        00:00:01 avnpython /awips/adapt/avnfps/3.2/py/avndrs.py -d
-n px2f
fxa      11508 11280 11508  0   10 Mar  01 ?        00:00:00 avnpython /awips/adapt/avnfps/3.2/py/avndis.py -d
-n px2f
fxa      11508 11280 11511  0   10 Mar  01 ?        00:00:13 avnpython /awips/adapt/avnfps/3.2/py/avndis.py -d
-n px2f
fxa      11508 11280 11512  0   10 Mar  01 ?        00:00:15 avnpython /awips/adapt/avnfps/3.2/py/avndis.py -d
-n px2f
fxa      11508 11280 11513  0   10 Mar  01 ?        00:01:41 avnpython /awips/adapt/avnfps/3.2/py/avndis.py -d
-n px2f
fxa      11508 11280 11516  0   10 Mar  01 ?        00:00:02 avnpython /awips/adapt/avnfps/3.2/py/avndis.py -d
-n px2f
fxa      11508 11280 11517  0   10 Mar  01 ?        00:00:01 avnpython /awips/adapt/avnfps/3.2/py/avndis.py -d
-n px2f
fxa      11508 11280 11520  0   10 Mar  01 ?        00:00:00 avnpython /awips/adapt/avnfps/3.2/py/avndis.py -d
-n px2f
fxa      11508 11280 11526  0   10 Mar  01 ?        00:00:11 avnpython /awips/adapt/avnfps/3.2/py/avndis.py -d
-n px2f
fxa      11508 11280 11527  0   10 Mar  01 ?        00:00:43 avnpython /awips/adapt/avnfps/3.2/py/avndis.py -d
-n px2f
fxa      11508 11280 11532  0   10 Mar  01 ?        00:00:15 avnpython /awips/adapt/avnfps/3.2/py/avndis.py -d
-n px2f
fxa      11509 11280 11509  0    1 Mar  01 ?        00:00:10 avnpython /awips/adapt/avnfps/3.2/py/avnxs.py -d
-n px2f
```

Note

Red Hat Enterprise 4™ threads are listed differently by **ps**. The -H (process hierarchy) option cannot be used with the thread (m, L or T) request.

3.2. Stopping AvnFPS Servers

Normally you should use the script **remoteServers.sh**:

```
lx2-wfo:96: /awips/adapt/avnfps/bin/remoteServers.sh stop
```

Another option is to log on to the host(s) where AvnFPS servers run and use the **avnkill** command. This command is also located in `/awips/adapt/avnfps/bin` directory. The **avnkill** first issues a `SIGTERM` signal, then it waits 30 seconds for all threads to terminate gracefully. If the threads do not exit, **avnkill** issues a `SIGKILL` signal to those processes that continue to run. Yet another option is to send a `SIGTERM` signal to **avninit**. For example listed in the previous section, to stop **avninit**, the command would be:

```
px2-wfo:4: kill 11280
```

This should stop all the servers since **avninit** sends `SIGTERM` to all its child processes before quitting.

Note

When **avninit** stops, for whatever reason, the existing servers are stopped as well.

Under special circumstances, (e.g., a hung server), you may want to stop and start a single server. Since **avninit** monitors the servers and starts those which are not running, the only action needed is killing the hung process. Bear in mind that the servers are threaded and only the top-level thread accepts signals.

For the **ps** listing in the previous section, the hierarchy is represented by order. For instance, to restart the **avndrs** server, find the first occurrence of avndrs in the output above, and type:

```
px2-wfo:4: kill 11500
```

This will kill process 11500 and its threads. Once the **avndrs** server exits, **avninit** will start a new **avndrs** server instance.

4. Avnsetup

avnsetup is a graphical user interface (GUI) which can be used to configure most features of AvnFPS. In most cases, the data displayed and modified in **avnsetup** are contained in the various files described in Section 2, “Configuring AvnFPS” [].

avnsetup main GUI

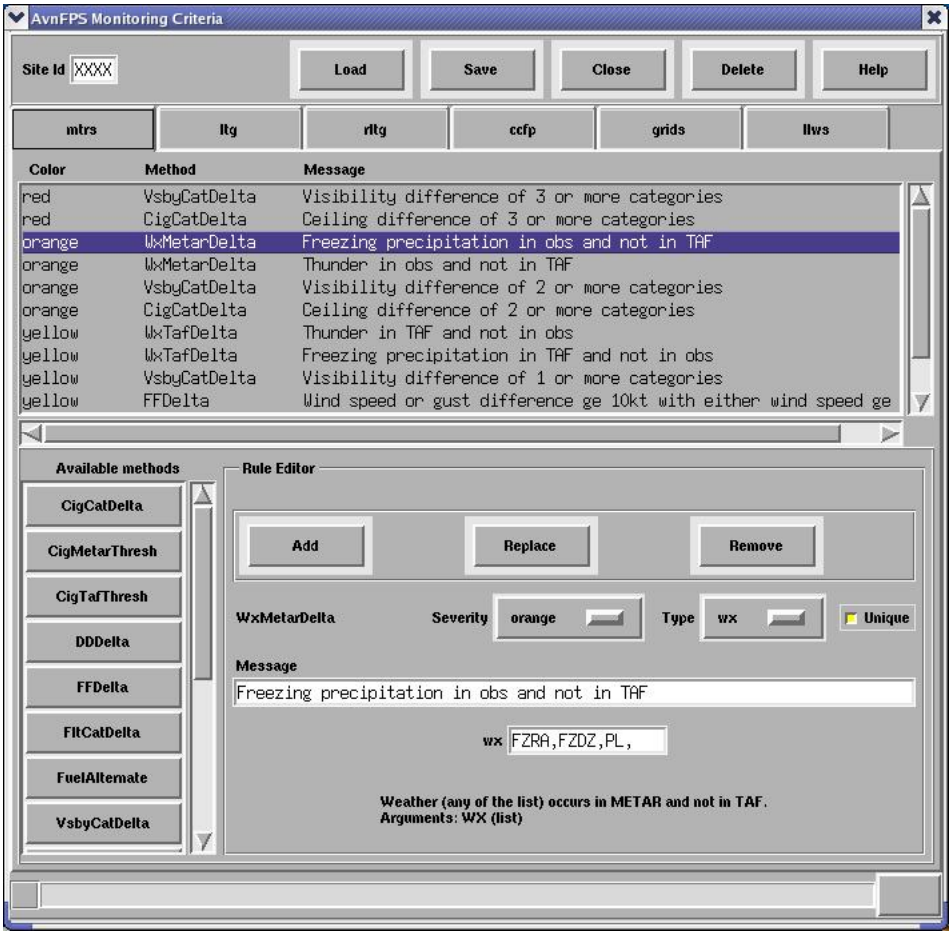
This section describes tasks that can be accomplished with this program.

Important

The **AvnWatch** GUI is required to be restarted for changes made using the following setup GUIs, except the Trigger Editor, to take affect. The reason for this, is that the underlying files, Section 2, “Configuring AvnFPS” [], are read once into memory on startup.

4.1. Editing Monitoring Rules

From the **avnsetup** main GUI, select the **Monitoring Rules** button. The Monitoring Criteria Editor will be shown.



Monitoring Criteria Editor

The Monitoring Criteria Editor contains several pages, one for each monitored data source. Pages are selected via “recipe tabs” near the top of the menu. The labels on the tabs refer to the following data sources:

- Rules for monitoring METARs (mtrs)
- Rules for monitoring lightning observations (ltg)
- Rules for monitoring lightning nowcasts (rltg)
- Rules for monitoring Colaborative Convective Forecast Product (ccfp)
- Rules for monitoring forecast grids (grids)
- Rules for monitoring low level wind shear (llws)

Each page has an identical layout: the top part lists currently defined rules, the bottom part contains a list of all available methods and the area where you can view, modify, add or delete a rule.

AvnFPS supports site-specific monitoring rules as well as a set of default rules. When the rule editor is started, the default rules are loaded. This is indicated by XXXX in the **Site Id** entry field.

To load the default set of rules:

- Enter the 'XXXX' into the box labeled **Site Id**.
- Press **Enter** or select the **Load** button.

To load the specific rules for a site:

- Enter the Station ID into the box labeled **Site Id**.
- Press **Enter** or select the **Load** button.

To create a new set of site-specific rules:

- Modify the rules using the instructions below.
- Enter the applicable Station ID into the box labeled **Site Id**.
- Select the **Save** button.

To remove site-specific rules:

- Enter the Station ID into the box labeled **Site Id**.
- Select the “recipe tab” for the data source for which you want to remove the rules.
- Select the **Delete** button.

The lower half of the Monitoring Criteria Editor allows you to view and modify each of the monitoring rules. The current set of rules is listed by color and message in a selectable list. The Rule Editor at the bottom of the menu shows details of the rule that is currently selected. To the left to the Rule Editor is a list of Available Methods. These methods are software techniques that compare the current forecast to the latest values received from the data source. Each method has its own set of arguments that configure its behavior.

To view the details of an existing rule:

- Select the rule from the list.

To modify an existing rule:

- Select the rule from the list.
- Use the Rule Editor to make the necessary changes. See Appendix B, *Monitoring Rules* [] for a description of editor options.
- Select **Replace**.
- Select **Save** to write changes to the configuration file.

To remove a rule:

- Select the rule from the list.
- Select **Remove**.

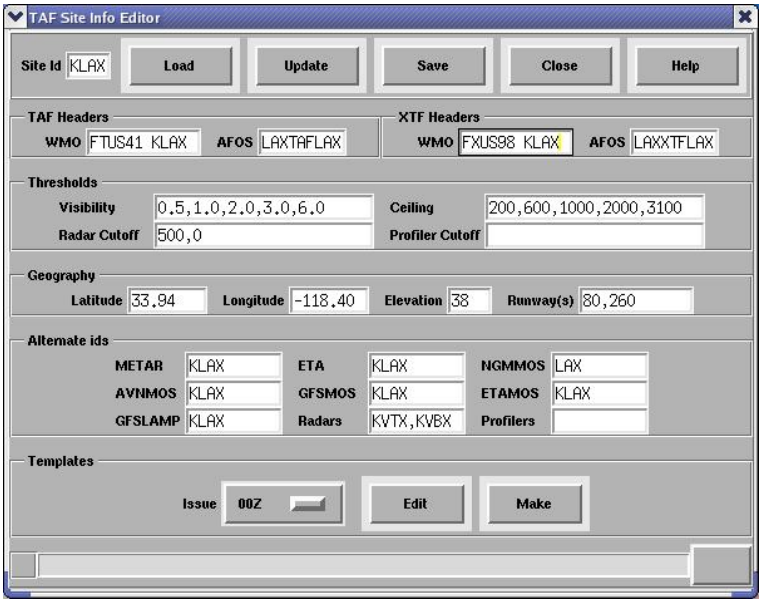
- Select **Save** to write changes to the configuration file.

To add a new rule:

- Select a method from the "Available Methods" list. Appendix B, *Monitoring Rules* [] provides detailed description. The Rule Editor will update with the appropriate fields. You can see method details by pointing the cursor to the respective button on the list.
- Use the Rule Editor to modify arguments as needed. Entering Message is no longer neccessary, if left blank, AvnFPS will generate one.
- Select **Add**.
- Select **Save** to write changes to the configuration file.

4.2. Editing TAF Site Information

From the **avnsetup** main GUI, select the **TAF SiteInfo** button. The TAF Site Info Editor will be shown.



TAF Site Info Editor

The TAF Site Info Editor is an entry form for data specific for a particular TAF site. Most of the fields can be initialized from data residing in AWIPS files or from built-in values. You must enter data for all sites before creating a product definition (that is, a "collective").

Site Id	TAF site ID
TAF WMO TAF AFOS	WMO header used to transmit forecast PIL used during transmission
XTF WMO XTF AFOS	WMO header used to transmit extended forecast (experimental) PIL used during transmission
Visibility	A comma separated list of visibilities (miles) defining categories used by the monitoring modules. Given the list $0 < V_1 < V_2 < ... < V_n$, the category k consists of visibilities V such that $V_{k-1} <= V < V_k$, k = 1,..., n+1 and $V_0 = 0$, $V_{n+1} = \#$.
Ceiling	A comma separated list of ceilings (feet) defining categories used by the monitoring modules. The categories are defined the same manner as above.
Radar Cutoff	A comma separated list of heights in meters defining lowest height to be used in computing LLWS using this radar. The length of the cutoff list must be equal to the number of radars listed in the 'Radar' text field below.
Profiler Cutoff	A comma separated list of heights in meters defining lowest height to be used in computing LLWS using this profiler. The length of list must be equal to the number of profilers listed in the 'Profiler' text field below.
Latitude	Site latitude, in degrees north

Longitude	Site latitude, in degrees east
Elevation	Site elevation, in meters
Runway(s)	A comma separated list of runway directions in degrees. Used to calculate cross wind.
METAR	METAR site id associated with the TAF, most likely the same as TAF. If more than one site is needed, separate entries by a comma.
ETA	The closest site id found in ETA BUFR message associated with the TAF.
NGMMOS	The closest site id found in NGM MOS message associated with the TAF.
AVNMOS	The closest site id found in AVN MOS message associated with the TAF.
GFSMOS	The closest site id found in GFS MOS message associated with the TAF.
ETAMOS	The closest site id found in ETA MOS message associated with the TAF.
GFSLAMP	The closest site id found in LAMP message associated with the TAF. Not yet available
Radars	A comma separated list of radars associated with the TAF. Can be empty.
Profilers	A comma separated list of profilers associated with the TAF. Can be empty.

To create data for a new TAF site:

- Enter the Station ID into the box labeled **Site Id**.
- Select the **Update** button. Most of the fields will be filled in. The headers are read from AWIPS file `afos2awips.txt`. Latidude, longitude and elevation (in meters) come from `metarStationInfo.txt`. Other values are built-in defaults.
- Modify displayed entries as needed. The **METAR** is mandatory, all others are optional. A simple validation test is performed when you type. The program will not allow you to enter obviously incorrect values, incomplete values are indicated by a pink background.
- Select the **Save** button.
- Create templates as described in the next section.

To modify TAF site data:

- Enter the Station ID into the box labeled **Site Id**.
- Press the **Load** button.
- Modify displayed entries as needed.
- Select the **Save** button.

4.3. Editing TAF Site Templates

The bottom part of TAF Site Editor provides tools to create and edit template files that can be used to initialize forecasts. There is a separate template for each forecast issue time. This allows to enter issue specific phrases such as `AMD NOT SKED AFT ...`.

To create TAF templates:

- Enter the Station ID into the box labeled **Site Id**.
- Select the **Make** button. This will create all four templates containing Station ID, issue and valid times. For example, a 06Z template for KPIT is:

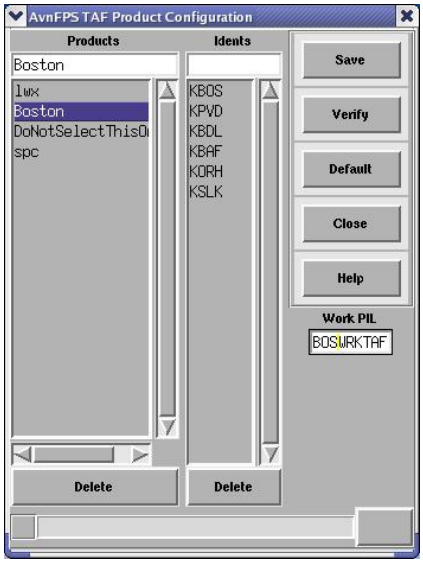
```
KPIT DD0520Z DD0606 =
```

To modify TAF template:

- Enter the Station ID into the box labeled **Site Id**.
- Select issue time from **Issue** menu and press the **Edit** to invoke a text editor. *Do not* press the **Make** button, as this will overwrite your current templates.

4.4. Editing TAF Product Definition

From the **avnsetup** main GUI, select **TAF Products** button. TAF Product Configuration Editor will be shown.



TAF Product Configuration Editor

The existing products will be displayed in the alphabetical order, with the default (if defined) on the top of the list. The **Idents** window displays list of all TAF sites in the product.

To add a product:

- Type the product name in the text window under the **Products** label and press **Enter**. An error message will be displayed that the product cannot be loaded, this is ok, as it's being created.
- Enter Site ID in the window under the **Idents** label. Hit **Enter** after each site.
- Enter PIL for work TAF in the window under the **Work PIL** label. This TAF product collective work product will be saved under this PIL in the text database.
- When finished, press **Save**. A new product definition file will be created.
- Press the **Verify** button to check whether all necessary files exist. Site info and templates must exist for all Site IDs listed in the product definition.

To modify an existing product:

- Select the product on the **Products** list.
- To add a new TAF site, type the ID in the window under the **Idents** label and hit **Enter**.
- To delete a site, select it on the **Idents** list and press the **Delete** button.
- Press **Save** to update product definition file.

To delete a product:

- Select the product on the **Products** list.

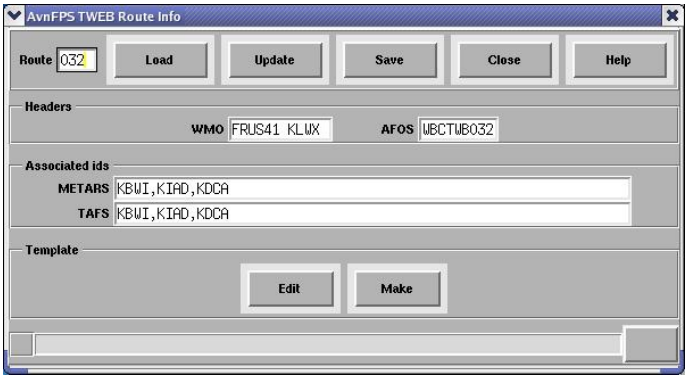
- Press **Delete**. A confirmation dialog will be displayed. Press **OK**, this will delete product definition file. TAF site files (info and templates) will not be removed.

To mark a product as the default:

- Select the product on the **Products** list,
- Press the **Default**. This will designate the selected product as default product.

4.5. Editing TWEB Route Information

From the **avnsetup** main GUI, select **TWEB Route Info** button. An empty TWEB Route Info Editor will be displayed.



TWEB Route Info Editor

The TWEB Route Info Editor is an entry form for data specific for a particular route or synopsis. You must enter data for all routes before creating a product definition (that is, a "collective").

Route	TWEB route number
WMO AFOS	WMO header used to transmit forecast PIL used during transmission
METARS TAFS	A comma separated list METAR site ids that should be displayed in the viewer window of TWEB Editor. A comma separated list TAF site ids that should be displayed in the viewer window of TWEB Editor.

To add TWEB route data:

- Enter the route number into the **Site Id** box.
- Select the **Update** to initialize WMO and AFOS headers from AWIPS file `afos2awips.txt`.
- Enter the METAR and TAF site ids you want to see in the TWEB Editor in the **Associated Ids** area. If you prepare TWEB product comprising more than one route, assure that the values you enter in those two fields are the same for each route listed in the product.
- Select the **Save** button.
- Create templates as described in the next section.
- Select the **Make** button in the **Templates** area. This will create basic template file.
- If you want to customize the template, press the **Edit** to invoke a text editor.

To modify TWEB route data:

- Enter the Route Number into the box labeled **Route**.
- Select the **Load** button.

- Modify displayed entries as needed.
- Select the **Save** button.

4.6. Editing TWEB Route Template

The bottom part of TAF Site Editor provides tools to create and edit template files that can be used to initialize forecasts.

To create TWEB template:

- Enter the Route Number into the box labeled **Route**.
- Select the **Make** button. This will template file containing phrase:

  NNN TWEB DDHHMM . ALL HGTS AGL EXC TOPS.

  where *NNN* is the route number.

To modify TWEB template:

- Enter the Route Number into the box labeled **Route**.
- Press the **Edit** to invoke a text editor. *Do not* press the **Make** button, as this will overwrite your current template.

4.7. Editing TWEB Product Definition

From the **avnsetup** main GUI, select **TWEB Products** button. TWB Product Configuration Editor will be shown.

TWEB Product Configuration Editor

Existing products are displayed in the alphabetical order, with the default (if defined) on the top of the list. The **Idents** window displays list of all TWEB routes in the product.

To add a product:

- Type the product name in the text window under the **Products** label and hit **Enter**. An error message will be displayed that the product cannot be loaded, this is ok.
- Enter site id in the window under the **Idents** label. Hit **Enter** after each site.
- Enter PIL for work TWB in the window under the **Work PIL** label. This TWB collective work product will be saved under this PIL in the text database.
- When finished, press **Save**. A new product definition file will be created.
- Press the **Verify** button to check whether all necessary files exist. Route info and the template must exist for all routes listed in the product definition.

To modify an existing product:

- Select the product on the **Products** list.
- Select the product on the list. To add a new TWEB route, type the route number in the window under the **Idents** label and hit **Enter**.
- To delete a route, select it on the **Idents** list and press **Delete** button.
- Press **Save** to update product definition file.

To delete a product:

- Select the product on the **Products** list.
- Press **Delete**. A confirmation dialog will be displayed. Press **OK**, this will delete product definition file. TWEB route files (info and template) will not be removed.

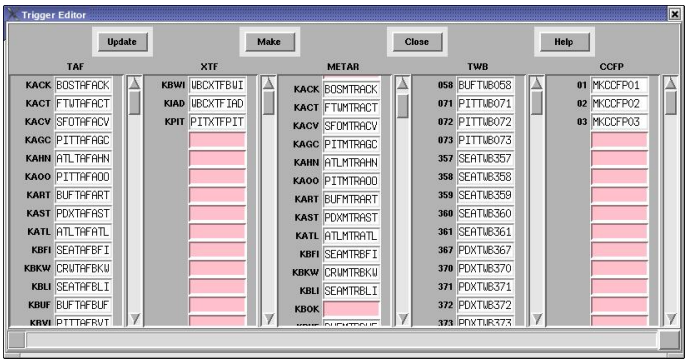
To mark a product as the default:

- Select the product on the **Products** list.
- Press the **Default**. This will designate the selected product as default product.

4.8. Creating Triggers

AvnFPS obtains TAFs, TWEBs, collaborative convective forecast products (CCFPs), and METARs from AWIPS Text Database (`fxatext`). As new versions of these products arrive, the `metarDecoder` can "trigger" another application--usually a script--to run. AvnFPS uses this triggering feature to retrieve TAF, TWEB, CCFP, and METAR products for monitoring as soon as they arrive in the database. The Trigger Editor manages the configuration of the `textdb` triggers needed to support AvnFPS. The current version creates trigger template, to be processed by AWIPS localization script and simultaneously updates the table `watchwarn` in `fxatext` database holding Postgres trigger scripts. So you do not have to run localization after creating the template file. However the file is still neccessary, as the localization process drops `watchwarn` table and recreates it from templates.

To start the Trigger Editor, select **Triggers** button. The Trigger Editor will be displayed.



Trigger Editor

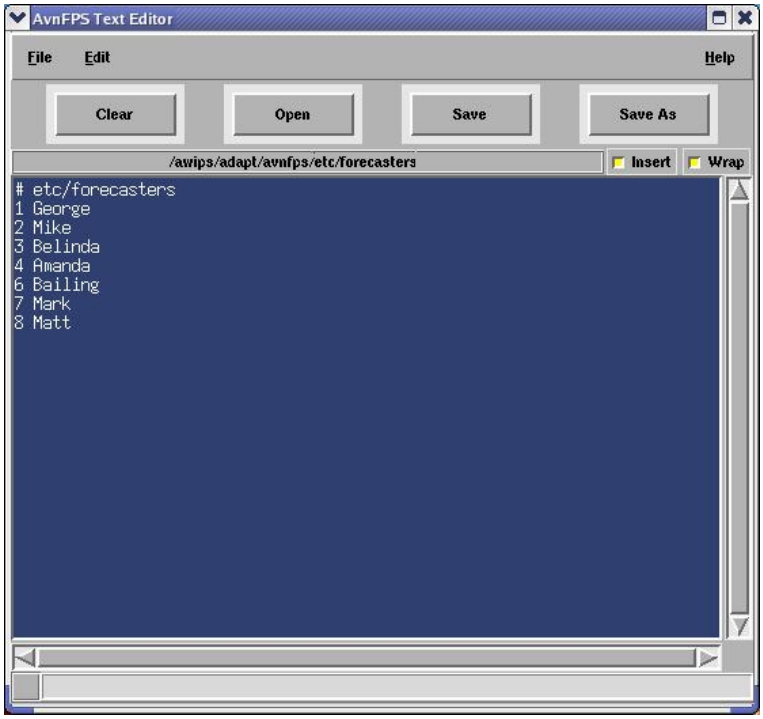
On startup, the Trigger Editor searches AvnFPS configuration files for all TAF sites, METAR sites, and TWEB routes that have been configured. (These are files stored in the directories `etc/tafs`, `etc/mtrs`, and `etc/twbs`.) The three columns: **TAF**, **METAR** and **TWEB** list those sites and routes. If there is a template file, `etc/triggerTemplate`, its entries are used to fill the corresponding PIL boxes. If the trigger template does not have an entry for an id, the relevant box will remain blank, with a pink background. The column **XTF** contains entries for extended (30 hour) TAFs. Unless your office issues such forecasts, it will be empty. Column **CCFP** should always have entries as shown above. Those text products are not site-specific.

To create AvnFPS triggers:

- Press the **Update** button. The program will try to determine missing PILs for TAFs and TWEBs from AWIPS file `afos2awips.txt`. METAR PILs are created by from corresponding TAF PILs by replacing `TAF` by `MTR`. The existing values *will not* be overwriten.
- Correct the entries and fill in empty fields.
- Press **Make** when finished. Empty (pink) entries will be skipped, if you have an invalid entry, an error message will be shown.

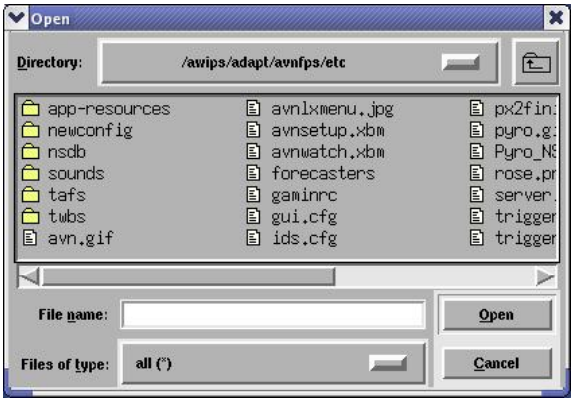
4.9. Using Text Editor

To edit remaining configuration files: `forecasters`, `logging.cfg`, `server.cfg`, `gui.cfg`, `ids.cfg`, `flt_cat.cfg`, `grp_taf.cfg` use the text editor. It is invoked by pressing **Text Edit** button.



Setup Editor Dialog

Press **Open**, you will get a file selection dialog listing content of directory `etc`. In this dialog, select a file to edit, then press **Open**.



File Selection Dialog

Press **Save** button when finished. If you want to create a new file, simply type its contents in the text window, and then use **Save As** button to save to disk.

You can also use the text editor to edit X resource configuration file. However, we suggest that you only modify the system-wide file which is used as a initial selection when a forecaster modifies his own resources through the graphical editor available in **AvnWatch**. Proceed as above, and in the file selection dialog, select the `app-resources` folder, then the file `X`. You may edit files for individual forecasters (those with names X.*N*),

however any comments you put there will be stripped by the graphical editor every time the forecaster modifies the resources from **AvnWatch**.

Tip

Font and color names may be cryptic. You may want to use the graphical editor in **AvnWatch** first to display color codes and available fonts.

5. Configuring IFPS to send gridded data to AvnFPS

With arrival of OB7.1 and DS decommissioning, it was decided not to port the IFPS applications, **ifpServerWatcher** and **avn_unldr** to use Postgres database resources. These two applications have been retired with OB7.1 and future builds. Their job was to monitor GFESuite's **ifpServer** for changes in the gridded database, and when notified of such changes, extract gridded data and place it in a format accessable to AvnFPS data ingest server. In their stead, a new **GFE** text formatter was created to do the job of accessing IFPS gridded database and writing the information to the `/data/adapt/avnfps/grids` directory.

The new **GFE** text formatter, **IFPS2AvnFPS.py** is run at least 10 times an hour via cron on the `px2f` machine. The only requirement for the new text formatter to run is that all TAFs *must have a **GFE** edit area defined* in the ifpServer database. We are working to remove this restriction in a later build. Fortunately, offices that have configured IFPS to send gridded data to AvnFPS most likely have already satisfied this requirement, so no additional work is required.

With IFPS software, the process of getting gridded data into AvnFPS was event-driven: triggered when a forecaster saved aviation-related parms in **GFE**. That has been replaced by scheduled-driven process using "fxa" cron. Ultimately, this new technique should be event-driven as well. This enhancement is planned for **ifpServer**.

6. Logging

AvnFPS has an extensive logging system, similar to collective logging in AWIPS. The amount of information written to log files and its format are configurable.

Each program has its own log file, one per day of the week. File naming is according to the scheme *`progname_DayOfWeek`*, so **avndis** will produce files `avndis_Mon` through `avndis_Sun`. Files are broken (old closed, new opened) at the first write attempt of each new day. This is different from AWIPS where a cron job breaks the log after 00Z.

The file `etc/logging.cfg` controls the location, format and log level for the log files. The format and log level is the same for all programs.

Note

The log directory is local to each server/workstation. The directory `/awips/adapt/avnfps/3.2/logs` points to `/data/logs/adapt/avnfps`, which resides on a local partition. This is consistent with the AWIPS standard:

```
lx2-wfo:46: ll /awips/adapt/avnfps/3.2/logs
lrwxrwxrwx    1 fxa      fxalpha        23 Oct  8 16:41 /awips/adapt/avnfps/3.2/logs ->
/data/logs/adapt/avnfps
```

Here is an example listing of a log file for **avndis**:

```
Log rollover at 00:00:01
2006-03-01 00:00:01,280 DEBUG [10517:126217136] TriggerThread: Processing PHLMTRMDT
2006-03-01 00:00:01,647 INFO  [10517:126217136] TextThreadP: Processed KMDT from SAUS70 KMDT 080000
2006-03-01 00:00:01,647 DEBUG [10517:126217136] TriggerThread: Done PHLMTRMDT
2006-03-01 00:00:01,649 DEBUG [10517:126217136] TriggerThread: Processing PITMTRDUJ
2006-03-01 00:00:01,656 DEBUG [10517:27261872] DataIngestServ: Msg: {'src': 'mtrs', 'ident': 'KMDT'}
2006-03-01 00:00:01,754 INFO  [10517:126217136] TextThreadP: Processed KDUJ from SAUS70 KDUJ 080000
2006-03-01 00:00:01,755 DEBUG [10517:126217136] TriggerThread: Done PITMTRDUJ
2006-03-01 00:00:01,756 DEBUG [10517:126217136] TriggerThread: Processing BUFMTRART
2006-03-01 00:00:01,981 DEBUG [10517:27261872] DataIngestServ: Msg: {'src': 'mtrs', 'ident': 'KDUJ'}
2006-03-01 00:00:01,989 INFO  [10517:126217136] TextThreadP: Processed KART from SAUS70 KART 080000
2006-03-01 00:00:01,990 DEBUG [10517:126217136] TriggerThread: Done BUFMTRART
2006-03-01 00:00:01,993 DEBUG [10517:27261872] DataIngestServ: Msg: {'src': 'mtrs', 'ident': 'KART'}
2006-03-01 00:00:05,289 DEBUG [10517:126217136] TriggerThread: Processing HFOMTRHNY
2006-03-01 00:00:05,576 INFO  [10517:101030832] LtgThread: Processing file
/data/fxa/point/binLightning/netcdf/20060228_2300
2006-03-01 00:00:05,629 INFO  [10517:126217136] TextThreadP: Processed PHNY from SAUS80 PHNY 080000
2006-03-01 00:00:05,631 DEBUG [10517:126217136] TriggerThread: Done HFOMTRHNY
2006-03-01 00:00:05,633 DEBUG [10517:126217136] TriggerThread: Processing BUFMTRROC
2006-03-01 00:00:06,020 DEBUG [10517:27261872] DataIngestServ: Msg: {'src': 'mtrs', 'ident': 'PHNY'}
```

Each line contains the following fields:

- Date and time of the event.
- Severity level (in caps).
- Process and thread number (in square brackets).
- Python module name which generated the message.
- Actual message content.

Some typical errors:

```
2005-07-08 02:00:48,843 INFO  [10517:84941744] MosData: Retrieved ngm data for BUF
2005-07-08 02:00:48,851 ERROR [10517:84941744] MosData: BVI not in /data/fxa/point/mos/NGM/netcdf/20050708_0000
(1)
2005-07-08 02:00:48,882 INFO  [10517:84941744] MosData: Retrieved ngm data for CHA

2005-07-08 05:32:52,556 DEBUG [10517:126217136] TriggerThread: Done MEMTAFTRI
2005-07-08 05:32:52,558 WARNING [10517:74451888] util: Daemon ** Exception during processing of request from
TCPConnection with ('xxx.xxx.xxx.xxx', 45051)
connected=1  type thread.error
(2)
--- traceback of this exception follows:
 ------------------------------------------------
 <thread.error> RAISED : can't start new thread

 Extended Stacktrace follows (most recent call last)
-------------------------------------------------
File "/awips/adapt/avnfps/Python-2.4.1/lib/python2.4/site-packages/Pyro/protocol.py", line (922), in
Daemon::connectionHandler
Source code:
    self.handleInvocation(conn)
File "/awips/adapt/avnfps/Python-2.4.1/lib/python2.4/site-packages/Pyro/core.py", line (695), in
Daemon::handleInvocation
Source code:
```

(1) This message indicates problem with TAF Site Info configuration file: data for KBVI is not distibuted for NGM MOS product. This does not cause any harm, but one can either select another site which is close enough, or leave the NGM MOS site id empty.

(2) Some messages provide detailed traceback to point out to the exact location of the code where the error occured. In this case, the error comes from Pyro code and it is relatively serious. The data ingest server hit the threshold of allowable memory consumption. **avndis** has this threshold set programatically to avoid clogging the whole operating system. In a situation like this one should restart **avndis** to avoid data loss. If the problem reoccurs, further investigation is required. (This example is contrived: the threshold was set too low during development).

7. Troubleshooting

Techniques for troubleshooting various aspects of AvnFPS are addressed in other sections of this manual. The purpose of this section is to gather some of the more likely fault scenarios and troubleshooting techniques into a single location.

7.1. Overall Server Health

A number of server processes keep AvnFPS running. A series of status lights on the TAF Monitor GUI reports the status of most of these processes. Keep alive messages are sent among the servers every 30 seconds to set the colors of these status lights. Section 1. of the System Administration Manual, "System Overview," contains a detailed description of each server and the host where it should be running. Server hosting can vary between AWIPS releases.

7.1.1. Diagnostic Tools

- If all the server status lights are red, check for
  - Failed name/event server (avnserver)
  - A network failure
- The command **`ps -efwH | grep avnpython`** lists server processes and any component threads. In this case, the operating system is Red Hat Enterprise 4 Indentation in the rightmost column indicates hierarchy. A sample listing follows:

```
px2-wfo:852: ps -efwH | grep avnpython
fxa      18615     1  0 Mar06 ?        00:00:00   avnpython /awips/adapt/avnfps/3.2/py/avninit.py px2f
fxa      18713 18615  0 Mar06 ?        00:00:33     avnpython /awips/adapt/avnfps/3.2/py/avnserver.py -d -n
px2f
fxa      18718 18615  0 Mar06 ?        00:00:01     avnpython /awips/adapt/avnfps/3.2/py/avndrs.py -d -n
px2f
fxa      18730 18615  0 Mar06 ?        00:01:08     avnpython /awips/adapt/avnfps/3.2/py/avndis.py -d -n
px2f
fxa      18735 18615  0 Mar06 ?        00:00:14     avnpython /awips/adapt/avnfps/3.2/py/avnxs.py -d -n
px2f

px2-wfo:ncfuser:853$
```

  Notes:
  - The command given above will shows the actual processes. To see the threads spawned by those processes use **`-efL`** flags to the ps command: **`ps -efL | grep avnpython`**.
  - Some comments follow:

PID	Comments	"Kill"-able?
18615	avninit daemon	Yes
18713	avnserver process	Yes
18718	Data Request Server (avndrs) process	Yes
18730	Data Ingest Server (avndis) process	Yes
18735	Transmit Server (avnxs) process	Yes

  - Threads of a process will not accept signals from a kill command. You must identify the process that owns the thread and kill that top-level process. Once a parent process terminates, all of its threads will terminate as well.
- The command **`netstat -a | grep 9090`** can be used to diagnose connections among the various servers. The name/event server uses port 9090 to disseminate information about the various servers that are up and running. Here's an example:

  Checking netstat on px2 …

```
px2-wfo:user:109: netstat -a | grep 9090
tcp        0      0 px2f-wfo:9090           px2f-wfo:55812          ESTABLISHED
tcp        0      0 px2f-wfo:9090           px2f-wfo:55819          ESTABLISHED
tcp        0      0 px2f-wfo:55819          px2f-wfo:9090           ESTABLISHED
tcp        0      0 px2f-wfo:55812          px2f-wfo:9090           ESTABLISHED
tcp        0      0 px2f-wfo:9090           lx3-wfo:40638           ESTABLISHED
```

Notes:

- The columns are Protocol, Receive Queue, Send Queue, Local Address, Foreign Address, and State.
- Six connections to and from the name/event server can be seen in the `px2` listing.
  - The first four established connections show connections between the servers on `px2` and the name/event server.
  - There is an additional connection to a process on `lx3`. Most likely, this is an instance of the **AvnWatch** GUI.

7.1.2. Stopping/Starting Servers

Avninit
: Starting the servers for AvnFPS is a little complicated because of the interelationships among them. The utility **avninit** was developed to handle these complexities. **avninit** is a persistent process that runs on `px2` or, in case of failover, `px1` and attempts to restart servers as needed. If you fix a file or network problem that is preventing a server from starting, **avninit** will attempt to restart the downed server. However, **avninit** will only attempt 10 restarts of a failed server in a one-hour span of time.

Environment

Important

All AvnFPS servers must run at user fxa; do not try to start them as root. The standard server startup scripts will check userid before launching the servers.

Avnkill
: The utility **avnkill** can be used to stop all AvnFPS servers that are running on a host. This includes **avninit**. **avnkill** will be persistent, sending interrupt signals first, then kill signals. It will also try to clean up ill-behaved child processes that refuse to die when the parent dies. Once all servers have stopped running, restart **avninit**, using the **remoteServers.sh start**

Bouncing Servers
: To "bounce" a Data Ingest, Data Request, or Transmisison Server, identify the top-level process associated with the application and use the kill command to stop it. Within a minute, **avninit** should launch a new instance.

7.1.3. Competing Servers

If instances of **avnserver** are running on both `px1f` and `px2f`, mayhem is almost guaranteed. Competing versions of avndis will run on `px1f` and `px2f` racing to access files in the `/awips/adapt/avnfps/data` directory tree. px1/px2 failover/failback procedures are designed to prevent this problem, but someone always seems to find a way... There should be one and only one name/event server running in the subnet.

7.2. Log Files

See Section 6: “Logging” (page 51) of the System Administration Manual, for complete information on logs. The following information is distilled from that section.

Log files for AvnFPS3.2 server processes are stored in the directory tree `/data/logs/adapt/avnfps`, local to the host computer. The names of the logs files are formed by the name of the application and the current day of the week (e. g., `avnmenu_Thu` and `avndis_Fri`). All applications use collective logging which means that log file entries from different instances of an application can be found, interleaved, in a single log file. The following screen capture shows sample directory listings:

GUI logs on a workstation
…

```
lx2-wfo:user:16: pwd
/data/logs/adapt/avnfps
lx2-wfo:user:17$ ls
avnclimate_Fri  avnmenu_Mon  avnmenu_Thu  avnqcstats_Tue  avnsetup_Thu  avnwatch_Mon  avnwatch_Thu
avnclimate_Wed  avnmenu_Sat  avnmenu_Tue  avnsetup_Mon    avnsetup_Tue  avnwatch_Sat  avnwatch_Tue
avnmenu_Fri     avnmenu_Sun  avnmenu_Wed  avnsetup_Sun    avnwatch_Fri  avnwatch_Sun  avnwatch_Wed
```

Server logs on px1/px2 …

```
px2-wfo:user:123: pwd
/data/logs/adapt/avnfps
px2-wfo:user:124: ls
avndrs_Fri  avndrs_Sun  avndrs_Wed   avninit_Sun  avninit_Wed    avnserver_Sat  avnserver_Tue
avndrs_Mon  avndrs_Thu  avninit_Fri  avninit_Thu  avnserver_Fri  avnserver_Sun  avnserver_Wed
avndrs_Sat  avndrs_Tue  avninit_Sat  avninit_Tue  avnserver_Mon  avnserver_Thu
```

It is possible for certain unexpected errors to go undetected in log files. One way to test against this possibility is to launch a GUI process from the command line and watch the standard error and standard output streams. Here is a command that will launch the AvnFPS startup menu from the command line:

```
lx2-wfo:ncfuser:19: /awips/adapt/avnfps/bin/avnstart.sh avnmenu
```

7.3. Data Ingest

Data Ingest Servers (instances of **avndis**) monitor events in AWIPS and decode various data as they become available. For some data types, the term decode means little more than copying a file. Once avndis has put the data into the AvnFPS directory tree, Data Request Servers (instances of **avndrs**) serve the data to the various GUIs.

7.3.1. Data Ingest Troubleshooting

- Log files for avndrs show data being delivered to GUIs as well as data access problems. See if GUIs are connecting to avndrs successfully and receiving data.
- Log files for avndis show data arriving and being decoded. Errors are rather easy to spot in here.

7.3.2. Specific hints for specific data types

Text (TAFs, METARs)	Database triggers deliver products to `/awips/adapt/avnfps/data/text`.
Text (TAFs, METARs; alternative)	As an alternative to triggers, the AWIPS acquisition server can be configured to copy arriving text products into `/data/fxa/point/avnfps/raw` which can be processed by **avndis**.
Lightning Observations	Directory `/data/fxa/point/binLightning/netcdf` is monitored for new and updated files.
Lightning Probability	Directory `/data/fxa/img/SBN/netCDF/LATLON/3hr/LTG` is monitored for new and updated files.
Low-Level Wind Shear	The following directories can be monitored for new files: • `/data/fxa/point/profiler/netcdf` • `/data/fxa/LDAD/profiler/netCDF`, • `/data/fxa/radar/`*`CCCC`*`/VWP`, where *`CCCC`* = radar-ID
IFPS Grids	IFPS controls the process that exports the data to `/awips/adapt/avnfps/data/grids` where it is monitored.

7.4. Climatology Files

Downloading and installing climatological data is described in Section 2.5: “Climatological Data” (page 27) of the System Administration Manual.

- URL for data for the new climatology files is http://www.mdl.nws.noaa.gov/~avnfps/data/hdf5 [http://www.mdl.nws.noaa.gov/~avnfps/data/hdf5/index.html]. Byte counts and checksums are posted for all files.
- Once the climatology files are placed in `/data/adapt/avnfps/climate` directory and the `etc/ids.cfg` updated, the new climatological applications, **WindRose**, **CVMonthly** and **CVTrends**, will be able access them.
- Verify checksums with the **md5sum** when files arrive on destination host. (The command is **md5sum** *`filename`*.) If a Windows™ host is used to transfer data, ensure that the OS makes no attempt to interpret the data. Under certain circumstances, Windows™ may try to perform end-of-line conversions on the files. This will be readily detected by a changed checksum value.

7.5. Product Transmission

Product transmission is covered in depth in Section 1.4: “Transmission Server” (page 7) of the System Administration Manual. Here are the recommended steps to follow:

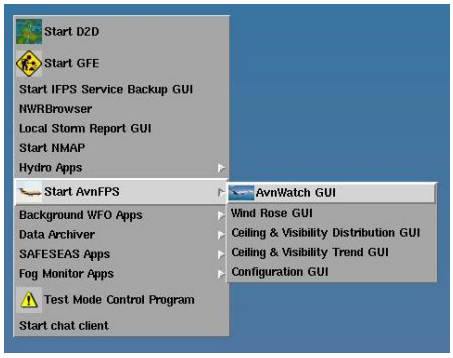
- Check log file for **avnxs**. If a forecast prepared by AvnFPS was written to the pending queue `/awips/adapt/avnfps/3.2/xmit/pending` and the transmission server attempted to acces the file, there should be a corresponding entry, starting with the word `SUCCESS` or `FAIL`*`NNN`*, where *`NNN`* is the code returned by the system call to **handleOUP.pl**.
- If **avnxs** indicates **handleOUP.pl** failure, proceed to investigate it's log file, `/data/logs/fxa/`*`yyyymmdd`*`/handleOUP`.

Chapter 2. User Manual

This manual is intended for forecasters using AvnFPS to monitor and prepare aviation forecasts.

1. Startup Menu

To display any of AvnFPS GUIs you must start with the AvnFPS startup menu via AWIPS popup menu.



AWIPS application launcher

When **AvnWatch** is started, you'll see a window as in the example here:

AvnFPS startup menu

The top list displays the names (or initials) of forecasters at your office. This name list permits individual customizations for things such as colors and fonts. The list at the bottom of the menu displays data server hosts than can be used to supply data to AvnFPS. For normal operations, the option named **local** should be selected.

To launch the TAF Monitor:

- Select your name/initials from the top list.
- Ensure **local** is highlighted on the lower list.
- Select **TAFs**button

To launch the Climatology Viewer:

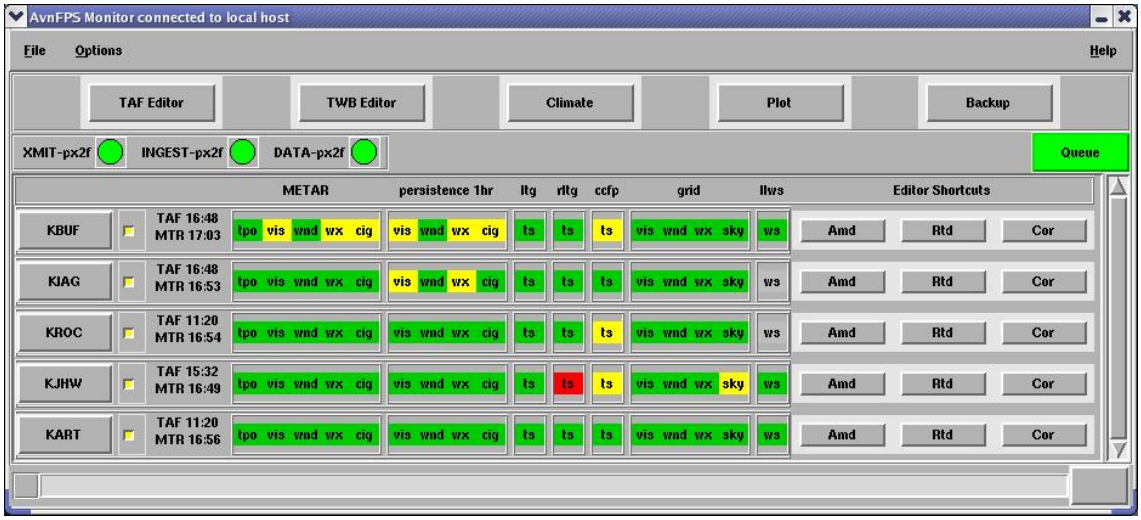
- Ensure **local** is highlighted on the lower list.
- Select **Climate** button

2. TAF Monitor

The AvnFPS TAF Monitor (aka **AvnWatch**) serves several functions.

- It displays information about the status of background services that are critical for correct functioning of AvnFPS.
- It displays results of monitoring TAF forecasts against observations and various guidance sources
- It provides access to the various editors that are part of AvnFPS.
- It allows you to customize the behavior of AvnFPS.

The image below shows the AvnFPS TAF Monitor.



AvnFPS Monitor

The TAF Monitor Window includes one row of buttons and labels for each TAF site of interest to your office. AvnFPS reviews the latest guidance, forecasts, and observations and reports the results in the color-coded portions of the window. Above the TAF Monitor Window is a row of Server Status Indicators. AvnFPS uses background processes, i.e. servers, running on one or more hosts to accomplish its tasks. Editor and Product Selection Buttons are arrayed on the Monitor as well. These buttons allow to quickly begin your routine and/or unscheduled product generation tasks. The Menu Bar allows you to access the user configuration tool for AvnWatch. Most of the user configuration items relate to the "look and feel" of AvnWatch on your workstation. Site-level configuration (add/remove a TAF site, change alert criteria) is generally accomplished with a different set of programs.

The message bar at the very bottom shows messages that do not require forecaster intervention (if an intervention is required, a popup dialog is displayed). When a text message is displayed for a short period of time, the label in the left lower corner flashes with color indicating severity of the message, which usually is one of the following (the colors are configurable): green: info, orange: warning, red: error. The time the message is shown depends on the severity and varies between 5 seconds for informational message to 20 seconds for system errors. You can view all messages that have been displayed by pressing the unlabeled button in the right lower corner.

2.1. TAF Monitor Window

The TAF Monitor Window contains rows of buttons and labels for all the TAF sites that have been configured for your office. Each item provides you with some information about the current TAF and how it compares to observations or guidance. A few of the items act as controls for the monitoring process.

GUI Elements

Site ID button
: These buttons are on the left-hand side of the TAF Window. These buttons invokes TAF Viewer. Any data that changes colors in the observation/guidance status (see below) causes the Site ID button to blink for 5 minutes (unless disabled). Problematic data, such as TAF containing errors are indicated by an orange background. Missing or non-decodable TAFs or METARs will make the background red. To stop the blinking, or to revert to the default background color, right click on the Site ID button.

Active Toggle Button
: The small square between the **Site ID** Button and the TAF/METAR Time Labels allows you to enable/disable monitoring of a TAF. When the box is selected (yellow), the current TAF for that site will be monitored.

TAF/MTR Time Labels.
: This label displays issuance times of the most recent TAF and METAR. The background color is used to notify the forecaster if one or both reports are late. When the last received report is older than the threshold value, the background changes to orange. The threshold value is 1hr 5min for METARs and 6hr 40 min for TAFs.

Observation/Guidance Status
: For each data source being compared to the TAF, one or more weather elements are monitored, using a set of rules configured by your office. Each element has a dedicated label. These are color coded, green if the element passed all the rules, gray if data were missing, and one of the following colors: pale green, yellow, orange, red and purple if one or more rules were violated. The color of the label suggests the severity of the violated rule with pale green being least severe and purple being most severe. The following data sources are available for monitoring:

  METAR
  The weather elements under the METAR heading represent a comparison of the current METAR observation with the current TAF. The five items are interpreted as follows:

tpo	TEMPO	An alert (generally yellow) is displayed for this item if the current TAF contains a TEMPO group and the conditions described in that TEMPO group have not been observed for more than half of the duration of that TEMPO period.
wnd	Wind	Alerts on discrepancies in wind speed, direction and runway cross wind component.
vsb	Visibility	Alerts on discrepancies in visibility.
wx	Weather	Alerts on discrepancies in precipitation and/or obstruction to vision.
cig	Ceiling	Alerts on discrepancies in ceiling.

  persistence *NUM*
  The weather elements under this heading compare the current TAF for *NUM* hours from now with the current METAR observation. The same rules are applied to the weather elements under this heading and the **METAR** heading. In effect, these columns assess the current TAF against the assumption that conditions in the current METAR observation will persist for *NUM* hours. The value of *NUM* is selectable: move the mouse over the 'persistence' label and right-click, a popup menu appears which allows you to select the number of hours. Weather elements and alerts are the same as for METAR, with the exception of the TEMPO check.

ltg
The weather element under this heading represents a comparison of the current TAF with the latest data from the lightning detection network.

ts	Number of strikes	This item will display alerts when lightning is observed near the TAF site and TS is not mentioned in the TAF.

rltg
The weather element under this heading represents a comparison of the current TAF with the latest lightning probability guidance. The lightning probabilities are created using algorithms developed for the System for Convective Analysis and Forecasting (SCAN), and they are generated hourly by the National Centers for Environmental Prediction (NCEP). See [[Kitzmiller et al.] [131]]. This guidance is issued every half hour and is valid for 3 hours.

ts	Lightning probability	This item will display alerts when lightning probabilities are high near the TAF site and TS is not mentioned in the TAF or vice versa.

ccfp
The Collaborative Convective Forecast Product is a graphical representation of expected thunderstorm occurance at 2-, 4-, and 6-hours after issuance time. The forecast is disseminated as a text product and stored in AWIPS fxatext database. AvnFPS decodes it, finds whether the monitored TAF sites fall within a convective area and alerts whenever convection parameters: coverage and confidence exceed customized thresholds and there is no thunderstorm in the relevant period(s) of the TAF. See [[CCFP] [131]] for description of the CCFP message.

cnv	Convection potential	This item will display alerts when convection is forecast, but TS is not mentioned in the TAF. The check is done for all 3 forecast times.

grid
The gridded forecasts prepared by the Interactive Forecast Preparation System (IFPS) to support the National Digital Forecast Database (See [[NDFD] [131]]). AvnFPS can be used, however, to monitor the consistency of the TAF with the IFPS gridded forecast. The weather elements under this heading compare the current TAF with the IFPS forecasts for the gridpoint nearest the TAF site.

Note

Remember that the data displayed in the IFPS Guidance Window are generally representative of a larger area (2.5x2.5km2 or 5x5km2, depending on GFE grid resolution) than the TAF.

The guidance prepared by GFE is split into 3 groups:

wnd	Wind	Alerts for discrpancies in wind speed and/or wind direction.
wx	Weather	Alerts for discrepancies in weather.
sky	Sky cover	Alerts for discrepancies in sky cover.

llws
The weather element under this heading represents a comparison of the current TAF with the latest wind profile data captured from WSR-88D radar products as well as the NOAA Profiler Network.

ws	Wind shear value	This item will display alerts when wind profile data indicate wind shear is present near the TAF site and wind shear is not mentioned in the TAF

Note

Colors displayed in the weather element labels, data sources and menus are configurable, see System Administration Manual, the section called "`etc/gui.cfg`" [13].

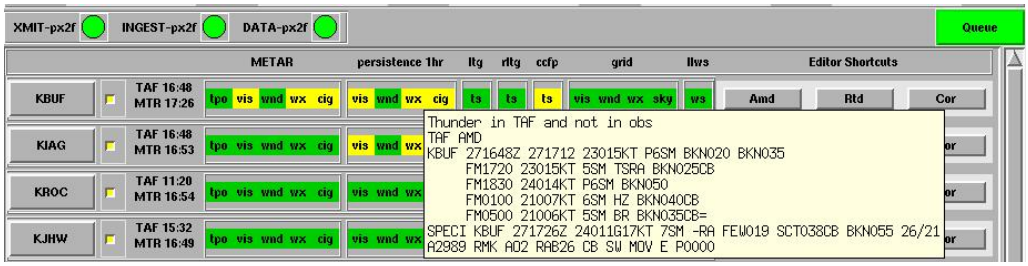
Editor Shortcuts

If you have configured AvnFPS to show the Editor Shortcuts (see Section 2.5, "Resource Editor" [73], the relevant resource is `*amdbuttons`), you will see three buttons to the right of the weather elements. These buttons allow you to quickly launch the TAF Editor to prepare amended, corrected, or routinely delayed forecasts.

Common Actions

To view monitoring details:

Hover over the weather element. A popup window will show displaying current TAF, observation/guidance and messages for all violated rules.



To view observations and guidance for a TAF site:

Select the ID button.

To stop the Site ID buttons from blinking:

Right-click the Site ID button.

To change the number of hours used in persistence monitoring:

Right-click on the persistence label, then select the number of hours from the pop-up menu that appears.

2.2. Server Status Indicators

The Server Status Indicators provide you with important information about AvnFPS' ability to ingest data and transmit products. If one or more of the Server Status Indicators is red, your Aviation focal point or ITO should investigate the cause. If data or ingest servers have failed, the monitoring functions of AvnFPS will be unreliable or fail completely. If the transmission server has failed, AvnFPS will not be able to issue products. Common causes for failures include:

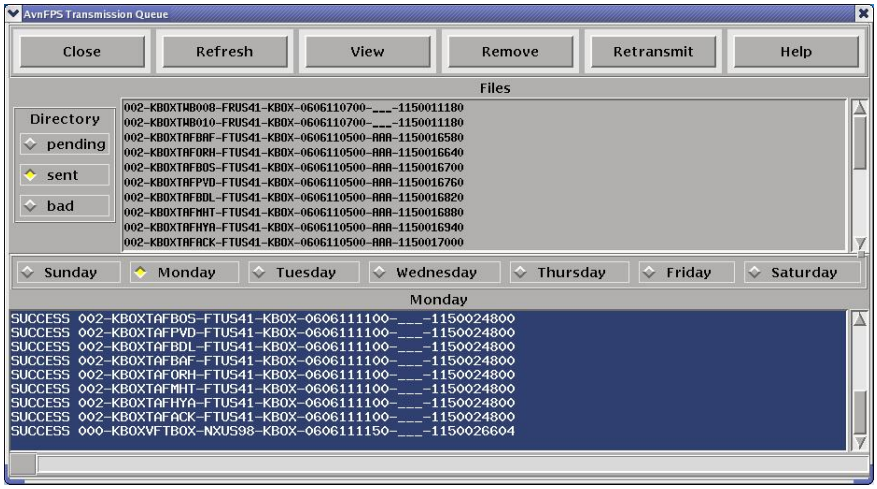
- A misconfiguration of system files,
- Computer or network maintenance work,
- Failures in system software or hardware.

If your aviation focal point and/or ITO is unavailable for assistance, the NCF should be contacted.

Tip

If all indicators are red, then its most likely there is a failure in the local network, and not a problem with AvnFPS per se. Once the network problem is resolved, the aviation focal point or ITO should restart the AvnFPS servers.

The **Queue** button to the right of the Server Status Indicators allows you to view and modify the forecast transmission queue, this is the list of products that are waiting for a specific time before they are transmitted. The background color of the **Queue** button reflects the status of the most recent transmission attempt: green for success, red for failure. Selecting the **Queue** button launches the AvnFPS Transmission Queue Dialog. A sample Transmission Queue Dialog is shown below:



Transmission Queue Dialog

The top window of the Transmission Queue Dialog, in this image, displays a list of files successfully sent out over the WAN. A set of radio buttons to the left side of window changes the display: you can see products waiting in the queue, 'pending', products successfully disseminated, 'sent', or the 'bad' files - either products that did not go out, or a rogue file that should not have been put in the queue. The bottom window displays the results of all transmission attempts for one day. The row of buttons labelled with the days of the weeks allow you to control the day for which the transmission log is displayed. AvnFPS maintains one week of logs.

The table below describes the actions of the buttons on the top row of the AvnFPS Transmission Queue Dialog:

Close
Closes queue viewer.

Refresh
Updates the information in the dialogs. This will not happen automatically; the viewer does not actively monitor log files.

View
Displays the content of the selected file.

Remove
Cancel the transmission of the selected forecast, waiting in the pending queue. Actually removes the selected file.

Retransmit
Schedules a forecast for retransmission. The issue, vaid and header times are *not* updated. Actually moves the selected file from the `xmit/sent` directory to the `xmit/pending` directory and updates the transmission time.

Help
Displays short help in a text window.

Note

If you want to make changes to the forecast in the TAF or TWEB editors, just reload it, the editor will pull out the one waiting in the queue.

Originally there was a hardwired time interval that a product could remain in the pending queue before being moved to the 'bad' queue. With AvnFPS3.2, the time interval is configurable. For more details on both of these features, see Section 5, “Forecast Transmission” [94].

2.3. Editor and Product Selection Buttons

Just below the menu bar there are two or three buttons.

TAF Editor
This button invokes TAF Editor.

TWB Editor
This button invokes TWEB Editor. If your office does not issue TWEBs, it may not be shown.

Backup
The **Backup** button can used to monitor a different set of TAFs on-the-fly. This can be useful when backing up other WFOs' TAFs. In some cases, the number of products may be small enough to configure a single instance of AvnFPS to monitor and edit all products issued by both WFOs. When you select the **Backup** button, the Product Selection Dialog launches:

Product Selection

Tip

If you want to select more than one item from the list, either press and drag the mouse (this works for consecutive items), or click at the items while pressing the **Ctrl** key.

2.4. Menu Bar

File

File Menu

Check Now (**Alt-f-c**)
Normally the checks are done upon reception on notification from **avnserver** (and every 30 minutes, to deal with TEMPO group verification). This option forces an immediate check of all TAFs versus current observations and guidance sources.

Restart (**Alt-f-e**)
This option shuts down the current instance of AvnFPS and restarts it, like after making a change to your X resource file. The restart option preserves the selection of monitored product(s). You also have a chance to choose a different forecaster id.

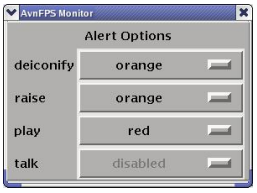
Quit (**Alt-f-q**)
Closes the application.

Options

Options Menu

Setup (**Alt-o-s**)
Can be used to edit forecaster specific configuration file `/etc/app-resources/X.N` where *N* is the forecaster ID. See resource editor below.

Alert (**Alt-o-a**)
Used to select alert criteria when the monitor detects a condition requiring forecaster's action. Alert pops up a dialogue in which you may select severity level for activating particular option.



Alert Dialog

Blink (**Alt-o-b**)

If selected, the color circle label will blink on new notification about changing weather conditions.

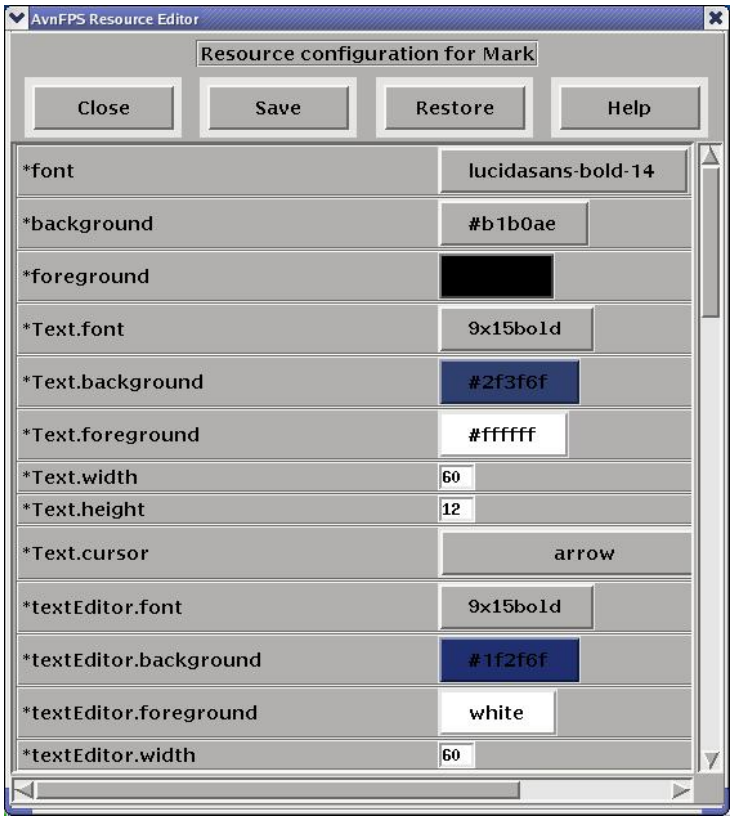
Help

Help Menu

About (**Alt-h-a**)
Provides version number, and contact information.

Usage (**Alt-h-u**)
Displays online help window.

2.5. Resource Editor



Resource Editor Dialog

This dialogue can be used to edit resource configuration files for individual forecasters. Within the scrolled window there is a list of configuration resources. Each frame contains name of the resource, you can display a short description in a balloon pop-up window by pointing mouse cursor at the name. Value of the resource is displayed on the right. Depending on the resource type it is:

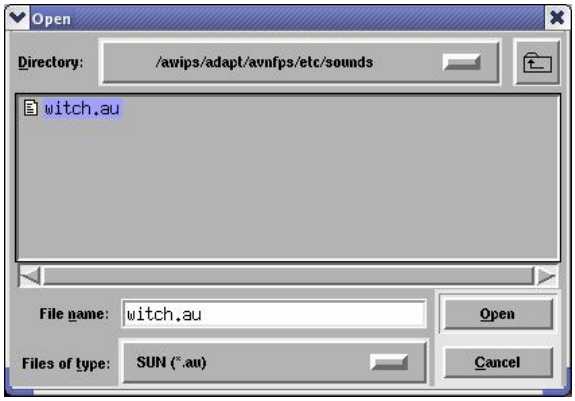
Toggle Button
Used for Boolean values: yes or no (1 or 0 in the configuration file).

Option Menu
Allows selection from predefined values.

Entry
Allows to type in values, such as width and height

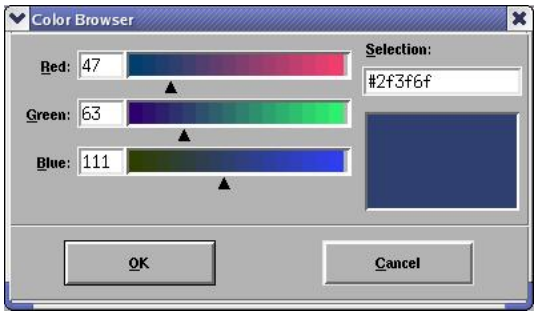
Button
Invokes dialogues specific for a given resource. This can be:

File Selection Dialog
Used to select audio file for alerts, resource `*playFile`



File Selection

Color Chooser Dialog
Alows to set various colors.



Color Chooser

Font Chooser Dialog
Selects fonts froma list of all fonts available on the system.

Font Chooser

The toggle **Short Names** resticts the list to font aliases. This significantly shortens the length of displayed fonts and is generally sufficient.

3. TAF Editor

The TAF editor is a text editor with all the standard capabilities such as cut, copy, paste, find and replace, auto backup. Forecast specific functionality involves syntax checking, climate and current weather consistency, and if configured, airport impact checks. It can also be invoked as a forecast viewer, where inconsistencies between forecast and the recent observations are highlighted.

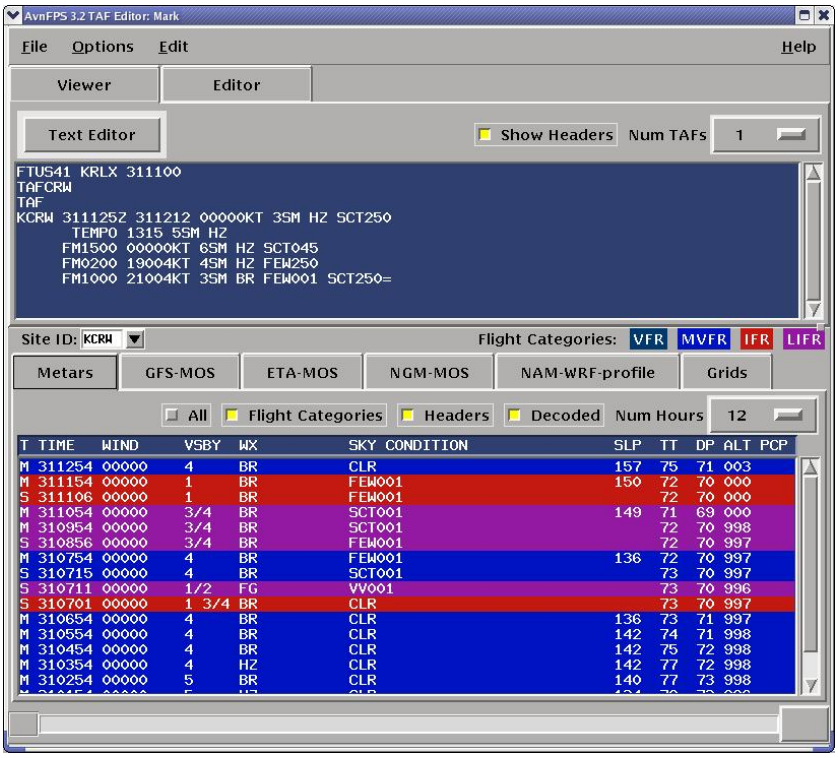
3.1. Starting the TAF Editor

The editor is invoked from the TAF Monitoring GUI. There are 3 ways to do it.

- Presing the **Taf Edit** button will display the editor in the edit mode without loading forecasts. Normally, you would use this method when preparing your routine TAFs.
- Use one of the Editor Shortcuts for the selected TAF site on the right-hand site of the TAF window. This will display the editor in the edit mode, with the most recent valid TAF loaded.
- Press the site id button for the selected TAF site. This will display the editor in the viewer mode.

The upper portion of the TAF Editor can be used to either view or edit TAFs. Two "recipe tabs" near the top of the editor allow you to choose between these two modes.

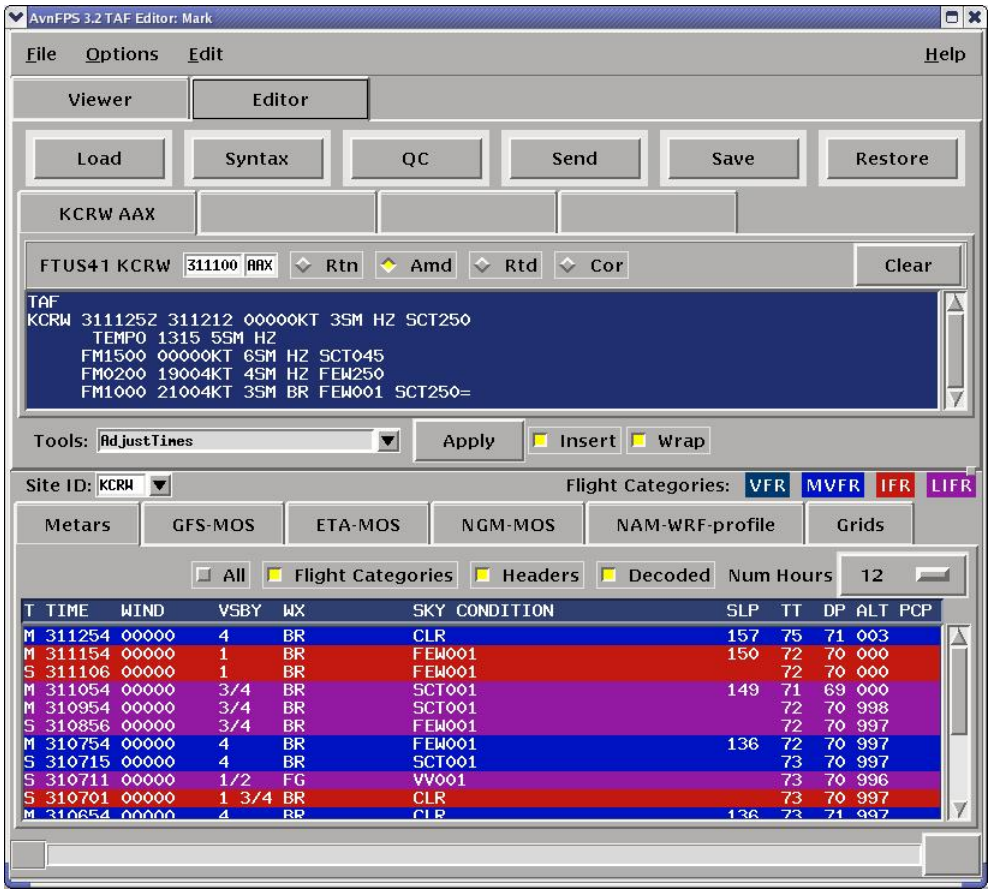
3.2. Using the TAF Viewer



TAF Editor - Viewer Mode

The TAF Viewer Window shows one or more TAF products issued for a site. If the lower portion of the TAF Editor is set to show METARs, and the TAF disagrees with observations for that site, then portions of the TAF and the observation are highlighted to show the discreptancy. The **Text Editor** button will change the window to the TAF Editor Window and load the most recent TAF into a new tab. The **Show Headers** toggle will add/remove header information from the TAF products currently in the viewer. The menu labeled **Num TAFs** allows you to choose the number of TAF product versions that will appear in the viewer. The **Site ID** pull-down just below the viewer allows you to choose which TAFs, will appear in the viewer.

3.3. Using the TAF Editor



TAF Editor - Edit Mode

When the **Editor** recipe tab near the top of the TAF Editor is selected, the upper portion of GUI becomes a specialized TAF editor. Near the top of the Editor Window is a row of command buttons that perform several functions that can aid TAF preparation. Below the command buttons is a row of recipe tabs that allow you to edit a number of TAFs simultaneously. The text-editing window occupies roughly the upper half of the the GUI. Below the text window are options for running TAF Tools. TAF Tools are user defined functions that manipulate forecasts loaded in the TAF Editor.

To edit the text of a TAF that is loaded in the TAF Editor Window, click within the window. An insert cursor will appear at the point where you click. Edit text as you would with any application. Copy/paste functions are available from the Edit menu and from a right-click menu within the window. Search/replace functions are available from the Edit menu. Header information can be changed by using the menu items above the editing window. The current recipe tab can be cleared by selecting the **Clear** button.

Information on using the Command Buttons and TAF Tools can be found in the following sections.

3.3.1. Command Buttons

Load
Invokes the forecast selection dialogue.

Forecast Selection Dialog

To load TAFs in an TAF Editor window:

1. Select one or more products from the Products menu.
2. Select one or more sites from the Site menu.
3. Select an option from the **Initialize From** menu.
4. Select a **Forecast Type**.

Tip

If you want to select more than one item from the **product** or **Sites** list, either press and drag the mouse (this works for consecutive items), or click at the items while pressing the **Ctrl** key.

Options for initializing a TAF include the following:

latest	Use the most recent TAF otherwise use the template. Most useful for amended and corrected forecasts.
template	Use the template. Most useful for preparing routine forecasts.
merge	Merges the latest TAF (earlier portions of the forecast) and the template (end of the forecast). Most useful for sites with scheduled part time observations.

Important

If a forecast is waiting for transmission in `/awips/adapt/avnfps/3.2/xmit/pending` it will be loaded into editor, regardless of the menu choice, then deleted from the queue.

Syntax

Syntax stands for syntax quality control. When you press this button, the editor will attempt to decode all forecasts and reassemble them, assuring proper indentation and maximum line length. Any errors/warnings will be highlighted in the displayed bulletin. There are 3 levels of errors:

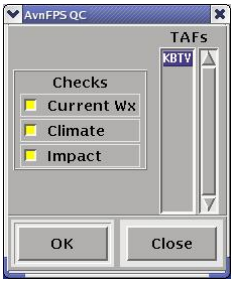
- Red - if the decoder cannot determine the meaning of particular word. This is a fatal error, the rest of the forecast (for a given site) is not decoded.
- Orange - this indicates an error as specified by NWSI 10-813 (shall not ...).
- Green - this indicates a warning as specified by NWSI 10-813 (should not ...).

If you press the left mouse button while pointing at the offending text, an explanatory message will be shown.

Syntax QC

QC

The **QC** combines several checks. Currently implemented are: Current Weather Check, Climate Quality Control and Impact Check. When you select this button with the left mouse button, all the checks will be performed on all forecasts in the editor window. Right button will display a QC Selection dialog, where you can pick individual TAFs and checks to perform.

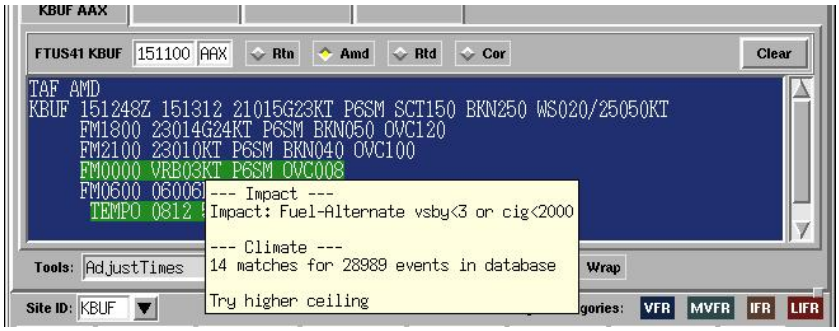


QC Selection Dialog

The weather check is intended for a quick check for changing conditions while forecasts are prepared, or when a routine issue forecast is prepared well in advance of the transmission window. If current observation and the weather in the first line of the forecast do not mach, the line will be highlighted.

Climate Quality Control is check that can assess the climatological frequency of the weather element combinations found in the TAF. It scans an archive of hourly and special observations, comparing combinations found in the TAF with observations available for that station. If a combination in the TAF has a low climatological frequency, an alert is displayed on the screen, alerting the forecaster to a climatologically rare event.
See Appendix C, *Climate Quality Control* [] for detailed description of the algorithm.

Impact check is a new feature in AvnFPS3.2. For each line of a TAF, a check is done against airport specific conditions. A message from Impact QC does not mean that the forecast is wrong just indicates that airport operations may be affected if the TAF is issued as is.



QC and Airport Impact Example

Save
Stores current TAF bulletin into a temporary file *tmp/CCCWRKTAF.type*, where *CCC* is your node id and *type* is the forecast type.

Restore
Restores bulletin previously stored in a temporary file *tmp/CCCWRKTAF.type*. *type* must match the type selected in the editor.

Send
Puts forecast in the transmission queue. In order to succeed, the forecast must successfully pass the Syntax check. In your resource file, there is an item *disalowSend, by default set to error. This means you can't inadverently send a forecast which does not follow the directive. If you have an error flagged, but are certain that the forecast is correct and either the software or directive is wrong, use **Clear Errors** before pressing **Send**. A confirmation dialog will be shown, where you will have to choose your name from the list. The corresponding forecaster ID number will be logged for forecast verification purposes.

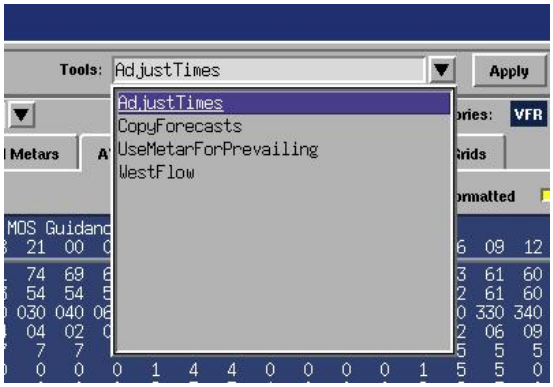
Send Dialog

You can also set the transmission time, this is to allow for sending routine forecasts at the end of transmission window, in case of rapidly changing weather conditions.

3.3.2. TAF Tools

TAF Tools are user-defined functions that manipulate forecasts loaded in the TAF Editor. Additional TAF Tools can be developed and implemented by programmers at your WFO, much like the locally-developed Smart Tools that can edit grids in the Graphical Forecast Editor. TAF Tools are written in Python, just like Smart Tools. TAF Tools can access just about any data source that appears in AvnFPS. This includes the TAF in the TAF Editor, other TAFs, METARs, and various forms of guidance.

A small set of TAF Tools are delivered with AvnFPS. These tools were developed for their instructional value as well as their utility. They are described below:



TAF Tools

Adjust Times
Modifies TAF by removing past periods, adjusts issue and valid times. If you use this tool before running syntax quality control, you can avoid error messages about invalid forecast periods.

Copy Forecasts
Allows for interactive copy of forecasts in a collective. The idea is that you prepare forecast for one site, and want to replicate weather conditions for some other TAF sites nearby. This tool can be an alternative for copy-and-paste.

Use METAR For Prevailing
Updates forecasts with the current observations. Also incorporates functionality of the **Adjust Times** tool.

West Flow
An example of a more sophisticated tool; it addresses a situation that commonly occurs in West Virginia. The tool creates forecasts for a number of other sites based on forecasts for KPKB and KHTS. The tool advects clouds and weather, adjusting cloud heights for elevation. Since the tool specifically access the WV TAF sites by name, it cannot be used “as is” by other WFOs.

3.4. Using the Guidance Viewer

The Guidance Viewer Window allows you to view various forms of guidance as you compose TAFs. Currently, AvnFPS can display up to seven data sources. These are Metars, GFS-MOS, ETA-MOS, NGM-MOS, NAM-WRF-Profile, IFPS Grids. Each source has a dedicated display window, which you select by clicking its tab.

All viewers have toggle **All**, if set, will display data for all sites associated with the currently monitored product. The toggle **Flight categories** will highlight periods according to the observed (METARs) and forecasted weather. Guidance is displayed in one of the 3 formats, depending on the toggle **Format**:

table	The display is similar to that of MOS text messages, with added optional display of probabilities for VIS and CIG categories (the MOS text bulletins contain only the best category).
long	The data is formatted as a TAF. There is one line per each forecast period (1 or 3 hours).
short	The data is formatted as a TAF. Forecast periods with similar weather are combined.

The toggle **Flight Categories** will higlight each forecast period/line depending on the flight category.

The algorithm to combine periods with similar weather is based on COMET [http://meted.ucar.edu/dlac/lesson3b] training module. Major rules applied are:

- Flight category: time periods are not combined in one TAF line unless it is configured by the users to merge by using `grp_taf.cfg`.
- Weather and wind condition changes: when there is a change (starting or ending) in weather conditions and wind changes, new line of TAF is always formed.
- Combining rules: Rules used to pack TAFs into more concise form. These rules are site configurable and are stored in `grp_taf.cfg`.

Metars

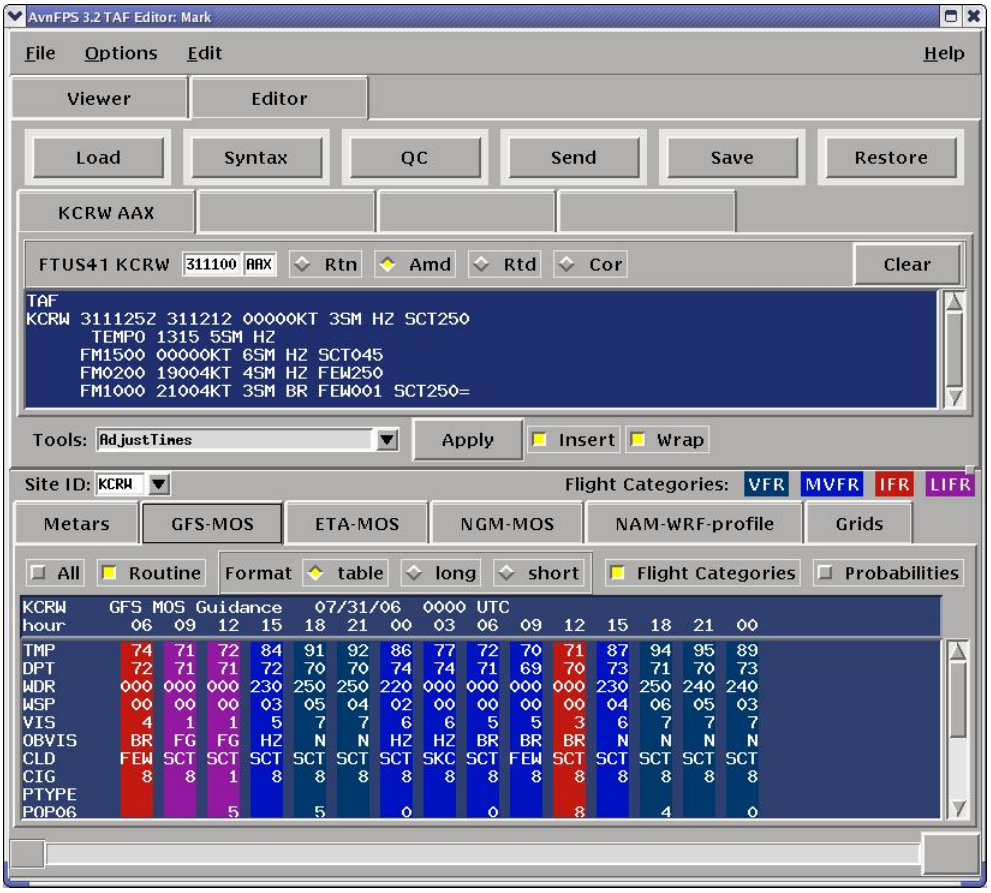
Displays METARs for a selected site. You can select the site from the **Site ID** list. The items on the list are set to the currently monitored product(s).

METAR Viewer

The number of displayed reports is controlled by the **Num Hours** menu. If there is a discrepancy between the TAF and the most recent METAR, the offending weather element will be highlighted. There are two display modes: text and decoded. In the text mode you may switch the display of WMO headers. In decoded mode there is an option to highlight reports according to the flight category. The toggle **Headers** is to display WMO headers. The toggle **All**, if set, will display all METARs for the selected product(s). It replaces **All METARs** found in the previous releases of AvnFPS.

GFS MOS Viewer

Displays GFS MOS guidance for a selected site. You can select the site from the **Site ID** list. The items on the list are set to the currently monitored product(s).



GFS MOS Viewer

The toggle **Routine** switches TAF valid time. **Probabilities** allows display of probability of individual ceiling and visibility categories. These data is not available in the MOS text products disseminated via SBN.

ETA MOS Viewer

Displays ETA MOS guidance for a selected site. The display format and control options are the same as for GFS MOS.

Note

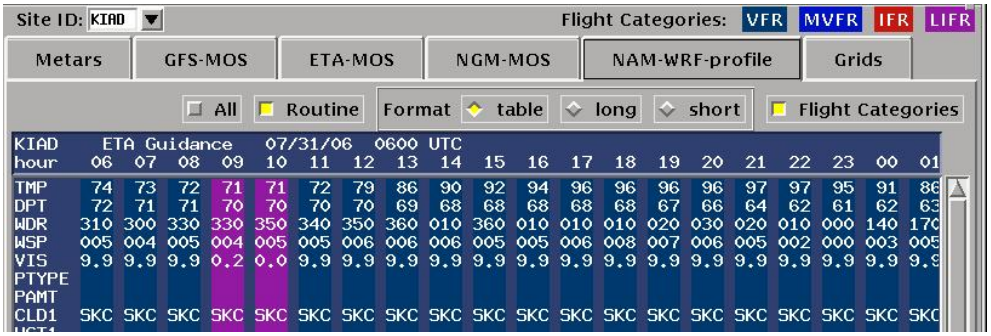
ETA MOS guidance will be discontinued after 12Z 31 December 2006.

NGM MOS Viewer

Displays NGM MOS guidance for a selected site. The display format and control options are the same as for GFS MOS.

NAM-WRF Profile Viewer

Displays model guidance from the NAM model for a selected site. You can select the site from the **Site ID** list. The items on the list are set to the currently monitored product(s).



NAM-WRF Profile Viewer

The display is similar as for GFS MOS guidance. There is no **Probabilities** button.

Note

Header information still refers to "ETA Guidance" in this display. This will be corrected in a later build.

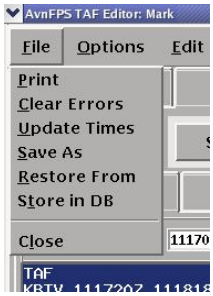
Grids Viewer

Displays data taken from IFPS grids for the gridpoint closest to the selected site. The display format and control options are similar to those for the GFS MOS Viewer, with the exception of **Probabilities** button.

3.5. Taf Editor Menu Bar

The Menu Bar allows you to access controls for many features of the TAF Editor. Entries in parentheses indicate keyboard shortcuts that will access that menu item.

File



File Menu

Print (**Alt-f-p**)
Launches the print dialogue. The dialogue displays the default print command: **lpr** and print on the default printer for your workstation. If you want to use a different printer, enter appropriate options (for example: **`lpr -Plp2`**).

Tip

You can use this dialogue to add content of the text window to fxatext database: enter **`textdb -w CCCNNNXXX`** with the appropriate AFOS PIL.

Clear Errors (**Alt-f-c**)
Clears error tags set by quality check actions. It can also be used to force transmission of forecasts that did not pass QC.

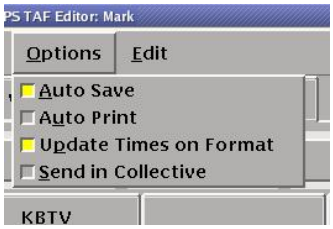
Update Times (**Alt-f-u**)
Updates issue and valid times in all forecasts loaded in the TAF Editor.

Save As (**Alt-f-s**)
Allows users to save edited forecast to a file. Invokes the file selection dialogue.

Restore From (**Alt-f-r**)
Allows users to restore forecast from a file. Invokes the file selection dialogue.

Store in DB (**Alt-f-t**)
Stores current TAF bulletin to fxatext database, under the AFOS PIL *CCCWRKTAF*, where *CCC* is your node id.

Options



Options Menu

Auto Save (**Alt-o-a**)
When selected, the content of the window is saved every 60 seconds to a backup file. If needed, the file can be restored by using the Files->**Restore** (**Alt-f-r**) The directory where the backup file is stored is `/awips/adapt/avnfps/3.1/tmp`. The file is named taf.*`host.kind`*, where *`kind`* is either taf or twb (for example taf.lx3). When a bulletin is reloaded, the file is renamed to taf.*`host.kind`*.prev.

Auto Print (**Alt-o-u**)
When selected, the forecast will be printed on the default printer when the **Send** button is pressed.

Update Times on Format (**Alt-o-p**)
When selected, issue and valid times in all forecasts will be updated (with values depending on the forecast type) when the **SynQC** button is pressed.

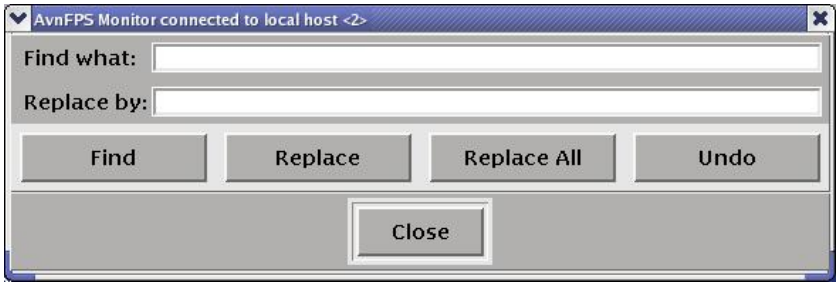
Send in Collective (**Alt-o-s**)
Normally, the bulletin is split before transmission, so that each TAF is sent separately. When this option is selected, the whole bulletin is transmitted as one file. This option is required at some OCONUS sites.

Edit

Edit Menu

Cut (**Alt-e-c**) , **Copy** (**Alt-e-o**) , **Paste** (**Alt-e-p**) , **Undo** (**Alt-e-u**) , **Redo** (**Alt-e-r**)
These options provide standard editing capabilities. Undo and Redo options have very large histories.

Find (**Alt-e-f**)



Find and Replace Dialog

This option invokes find and replace dialog with undo capability.

Tip

By pressing the right mouse button in the text area, you will get a popup menu, which also provides editing capabilities.

Editor Popup Menu

Help

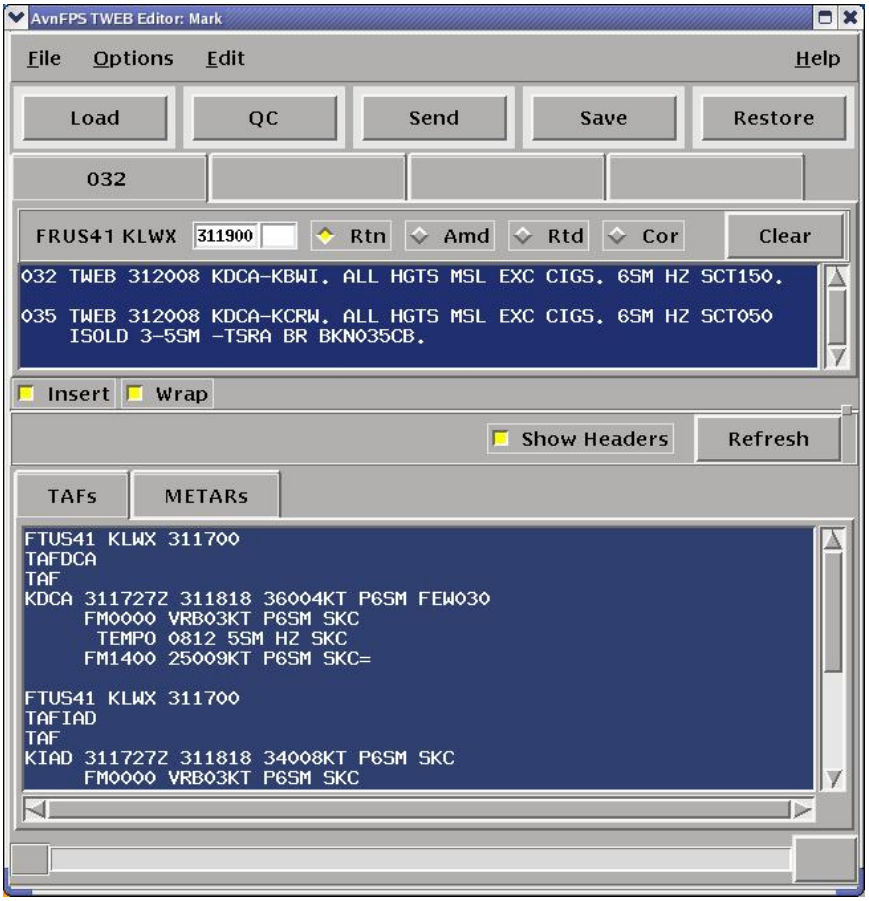
Key Bindings (**Alt-h-k**)
Displays key bindings valid in the text editor window. These are standard bindings as implemented in Tk. For complete list, see Appendix 1.

Usage (**Alt-h-u**)
Displays on-line help.

4. TWEB Editor

4.1. Starting the TWEB Editor

The editor is invoked from the monitoring GUI by selecting the **Twb Edit** button. This will display the editor without loading forecasts.



TWEB Editor

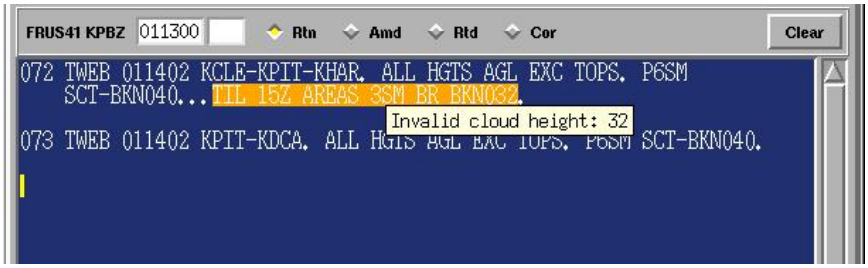
4.2. Using the TWEB Editor Window

The TWEB Editor is very similar to the TAF Editor. (See Section 3, “TAF Editor” [].) This section will highlight the differences between the two. Please refer to the appropriate sections of the TAF Editor documentation for detailed descriptions of TWEB Editor features.

The top part of the TWEB Editor consists of a text editor. (There is no TWEB viewer window.) The bottom part displays TAFs and METARs relevant to the edited forecast, as configured for your site.

Near the top of the Editor Window is a row of command buttons that perform several functions that can aid TWEB preparation. The command buttons have the same functionality as those in the TAF Editor. The **QC** button in the TWEB Editor provides functionality that is similar to the **Syn QC** button in the TAF Editor. Below the command buttons is a row of recipe tabs that allow you to edit a number of TWEBs simultaneously. Below the recipe tabs is a menu that contains a variety of header information about the TWEB that is being prepared in the editing window.

The quality control for TWEBs is very rudimentary and can be easily confused. The forecast is split into phrases, separated by a ".". The first phrase is checked for valid route, keyword `TWEB` and proper time format. The second phrase should contain either `NIL TWEB` or `ALL HGTS ...`. The subsequent phrases are checked for weather sequence and time. An attempt is made to check for valid visibility-obstruction to vision combination and for valid sequence of cloud layers. In case of an invalid entry, the whole relevant phrase is highlighted, and the explanatory error message will be displayed the same way as with the TAF syntax QC. There are 2 levels of errors, flagged in orange or green. No checks for valid abbreviation is performed. Proper formatting, such as maximum line length and indentation, is implemented.



TWEB Syntax QC

4.3. TAF/METAR Viewer Window

Analogous to the Guidance Viewer Window of the TAF Editor, the lower portion of the TWEB Editor can display METARs and TAFs that are relevant to the preparation of the current TWEB. Features in this window function the same way as they do in the TAF Editor.

Important

The TAFs and METARs are *not* updated by the editor. Use the **Refresh** button to update the display window.

4.4. Menu Bar

The options found on the Menu Bar of the TWEB Editor have the same functionality as their counterparts on the TAF Editor.

5. Forecast Transmission

Forecasts made within AvnFPS are sent by a transmission server. When you press the **OK** button in the **Send** [81] Dialog, the content of the forecast editor is split into separate TAFs/TWEBs which are then put in the transmission queue (that is, written to a file in the directory `xmit/pending`). The transmission servers checks the queue every *N* seconds. When the transmission time is reached, the forecast is disseminated. The transmission server will then notify all instances of AvnWatch whether the transmission succeeded. The AvnWatch's **Queue** button will flash for a while and the status of the transmission will be displayed in the Status Bar at the bottom of the Monitor. If the transmission fails, the **Queue** button will change its color from green to red. You can view the status by clicking on the button. This will display the Transmission Queue Dialog, see Section 2.2, “Server Status Indicators” [68]. To retransmit a forecast, select the **bad** option, highlight the failed entry (listed in the bottom window), and press **Retransmit**. The forecast will be moved back to the pending queue and the transmission server will attempt to send it again.

Note

The transmission server does not actually transmit the forecast. It simply invokes AWIPS utility **handleOUP.pl** and checks for the return status. If the forecasts are not transmitted, one should check **handleOUP.pl** log file, on the system where the server runs. Most likely it will be `px2`.

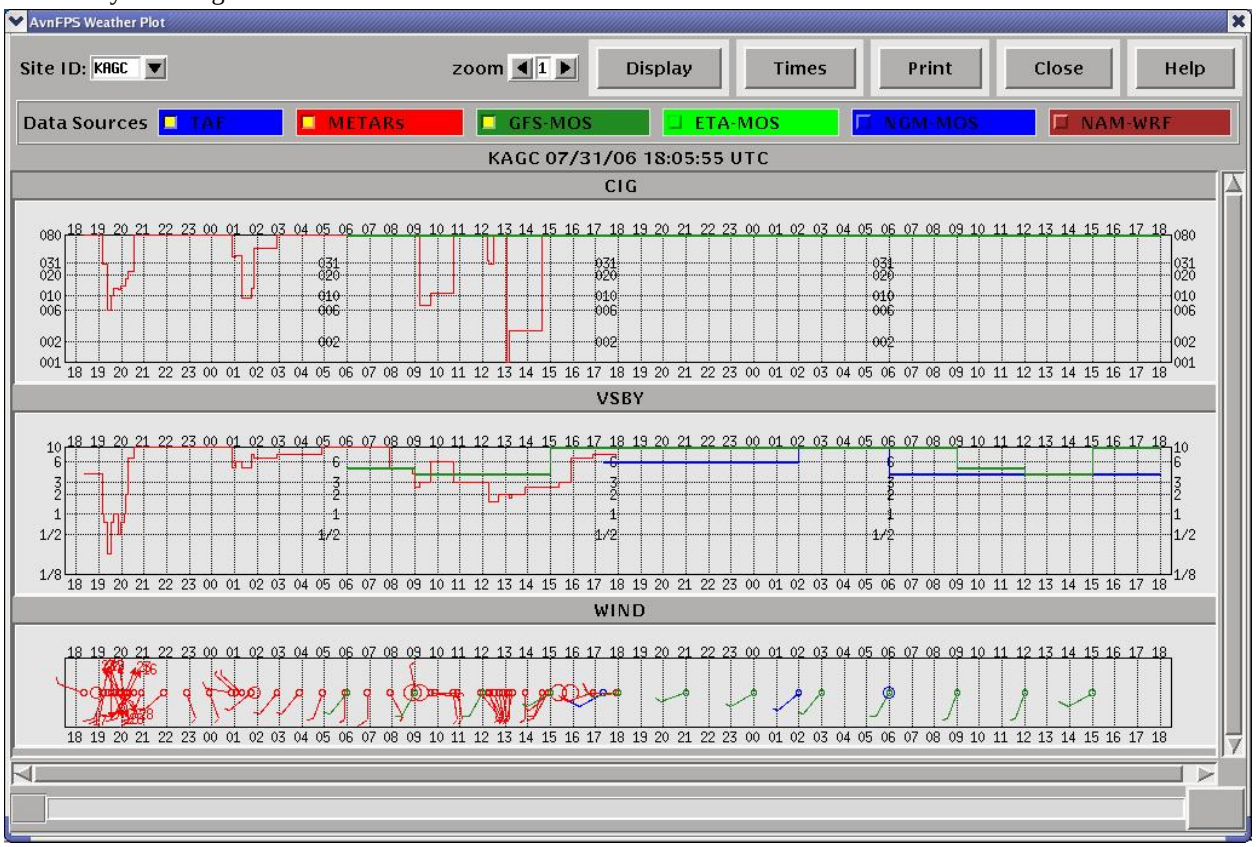
Important

If you scheduled a forecast to be transmitted at a future time, you can easily pull it from the queue if changing weather condition warrant its modification. Any time you load a forecast into the editor, it checks whether there is a similar TAF waiting in the pending queue. If so, it is pulled out from the queue and loaded into the editor. So, you always can modify a forecast, even if AvnFPS is restarted.

The transmission server collects information on forecasts sent by the WFO for verification purposes. A product with WMO ID header NXUS98 *CCCC*, where *CCCC* is your site id, is sent every 6 hours over the WAN where its collected as part of the Stats-On-Demand service.

6. Weather Plot

Weather Plot displays METARs, guidance and current and past forecasts in form of a time series. This dialog is invoked by selecting **Plot** in the TAF Monitor GUI.



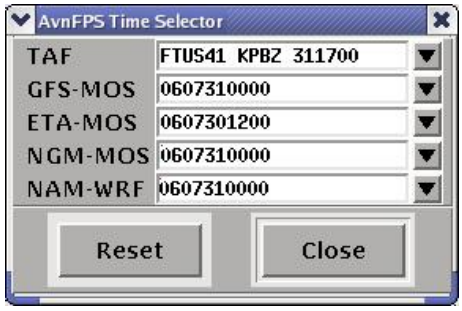
Weather Plot

Ceiling, visibility and winds are plotted as separate graphs. The horizontal axis in each graph represents time, the current time approximately centered in the display. The number of hours past and in the future is configurable. The ticks on the vertical axes for ceiling and visibility are the significant thresholds defined separately for each TAF site.

For each forecast period, ceiling and visibilities are represented by a tuple (low, high) values, and plotted as a shaded rectangle. Thus such a rectangle indicates existence of a TEMPO or PROB30 group. Only prevailing winds are plotted.

To plot data for a site, select the TAF id from the **Site ID** menu. This will display all data selected in the **Data Sources** area. Currently available data sources are: TAFs, METARs, ETA-MOS, NGM-MOS, GFS-MOS, NAM-WRF-Profile. Depending on the configuration at your WFO, only a subset may visible. However **TAF** and **METARs** buttons should always be present. By default, the most recent data is displayed.

If you want to plot previous forecast or guidance, select the **Times** button. A dialog displaying available data will be shown.



Time Selector

Click on the arrow on the right to show all available issuance times, select desired time, then press the **Display** in the main window. The **Reset** button resets all choices to the most recent time.

During changing weather conditions, the reporting frequency of METAR observations may be high, making the display, particularily wind plot, very "busy". In such case you may want to zoom the display, from the **zoom** menu select zoom factor, then press the **Display**. This button redraws the display and may be used when new observations are received, as the display is not automatically refreshed.

To save the graphs, use the **Print** button. By default, this will create a JPEG image in the `/awips/adapt/avnfps/3.2/tmp` directory. The WxPlot's **Print** command to run is located in the `etc/wxplot.cfg` file under the [print] tag. The '%s' wildcard--expanded to the identifier in the **Site ID** text field--in the command line string is optional. Thus, if you wanted the **Print** button to send the image directly to the printer, the command line string could be changed to this:

```
[print]
cmd=convert - -size 600x1000 jpg:- | lpr -o landscape
```

See Section 2.1, “Files in `etc`” [11] and refer to the `etc/wxplot.cfg` section for more details.

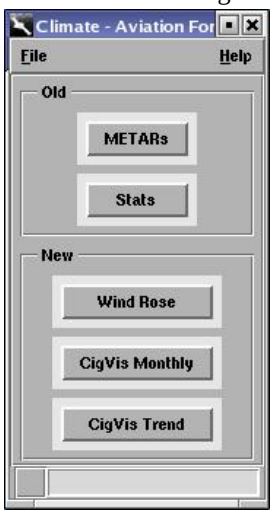
7. Climatology Tools

Beginning with AvnFPS 3.0, a climatology database was introduced as another guidance source to help forecasters prepare and QC their TAFs. The climatology database comprise of flat files, one for each TAF site. These files consist of NCDC yearly archives of surface observations, reformatted for fast indexing and retrieval. Since 3.0, forecaster feedback, and continued strong emphasis from the OCWWS Aviation Branch, has resulted in a second generation suite of climatology tools, introduced in AvnFPS 3.2. Description and use of the older tools are described in the following sub-sections. The newer tools are described in Section 8.

With AvnFPS 3.2, the older climatology tools and flat files are still available for use, however, they will be retired and removed in a later build.

7.1. Climatology Startup Menu

To start any of the climate tools, select the **Climate** button in the AvnFPS Startup Menu, see Section 1, “Startup Menu” []. The Climatology Startup Menu, contains two items. Select the **METARs** button to launch the Historical METAR Tool, and select the **Stats** button to launch the Climatological Statistics Tool.



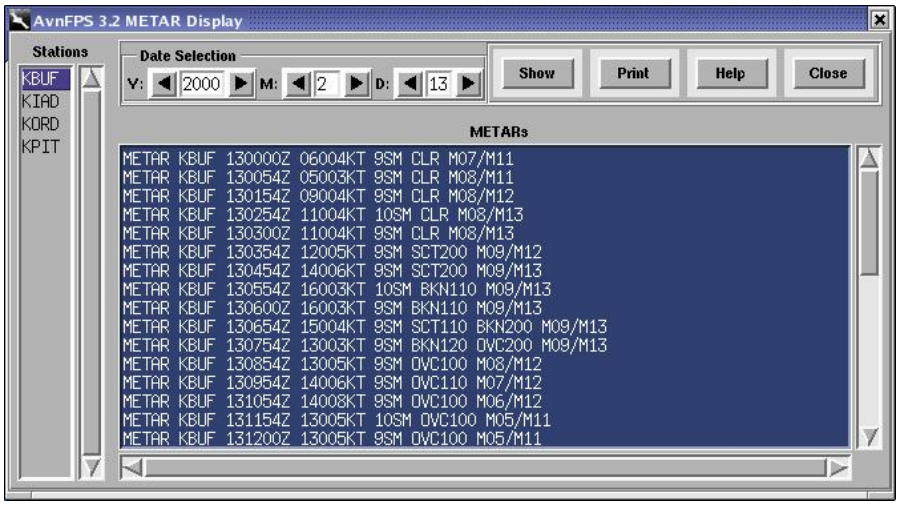
Climate Selection Menu

7.2. METAR Display

The Historical METAR Tool reconstructs hourly and special weather reports from the climatological database and renders them in the form of METARs. Each report contains as many of the following weather elements as are available: wind, visibility, significant weather, temperature, dew point, and 6- and 24-hr precipitation amount.

Important

The records in the database are merged from several data sources and the first records predate the era of METARs in the US. Thus the reports displayed are not true METARs.



METAR Viewer

To view observations for a particular day:

1. Select the desired station from the **Stations** list on the left side of the dialog.
2. Use the **Y**, **M**, **D** spinners to set the desired year, month, and day.
3. Select the **Show** button.

The window will be updated with the appropriate observations.

To print the set of observations:

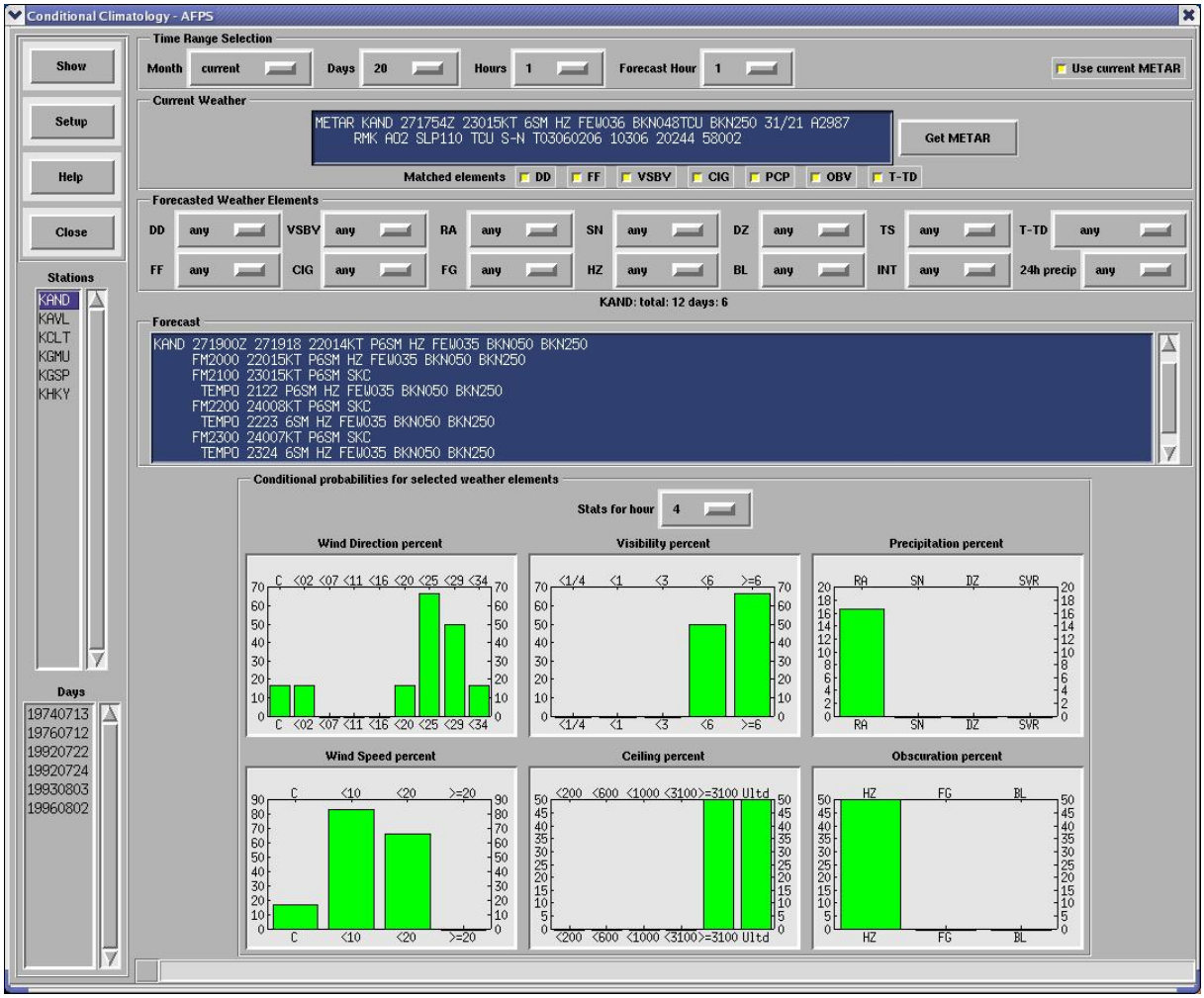
1. Select the **Print** button.

7.3. Climatological Statistics Tool

The Climatological Statistics Tool searches the climatological database for observations that match criteria you specify. The tool reports the results as a series of histograms. It also uses a time-lag matching technique to generate a guidance TAF based on observations found in the climatological database. There are two ways to control the search process, they are METAR-based and selection-based.

The Climatological Statistics Tool is divided into several sections. A set of buttons in the upper left corner control most of the database search capabilities. The **Time Range Selection** section controls time matching aspect during the database search. The contents of this section change depending on whether a search is METAR-based or selection-based. The **Current Weather** and the **Forecasted Weather Elements** sections specify the weather elements that will be used for the database search. The **Forecast**, **Conditional Probabilities** and **Days** sections show the search results. The IDs of stations that are available in the database are shown in the **Stations** section.

Below is an example of the Climatological Statistics Tool performing a METAR-based search:



Climate Statistics - 6 hour forecast

To perform a METAR-based search:

- Select the **Use current METAR** button. If it was off, the contents of the **Time Range Selection** section will change.

- Use the Stations list and the **Get METAR** button to load the current METAR for the site of interest. The latest METAR observation will appear in the METAR window. This window is editable, and the report itself can be modified. Use the **Matched elements** toggles to select which weather elements will be matched.
- Set the time range values for your search:
  - The **Month** button allows you to specify the month used in the search. Generally, the current month is sufficient.
  - The **Days** button specifies a range of days that will be used when searching. In the example, the date of the METAR is the 12th of the current month, and the day range is set to 20. That means the search will span 20 days either side of the 12th of the current month--a window of 41 days.
  - The **Hours** button specifies a range of hours that will be used when searching. In the example, the hour of the METAR is the 1900 UTC, and the hour range is set to 1. That means the search will span times from 1800 UTC to 2059 UTC.
  - If you expect a change in one or more weather elements, use the **Forecast Hour** button to specify the number of hours in the future when the change will occur. You must also use the **Forecasted Weather Elements** section to specify the changed value(s).
- If you selected a Forecast Hour in the **Time Range Selection** section, then use the option menus in the **Forecasted Weather Elements** section to choose proper value(s). Avoid selecting too many elements, or no matches will be found.
- Select the **Show** action button. The Forecast, Days, and Conditional probabilities sections will update after several seconds.

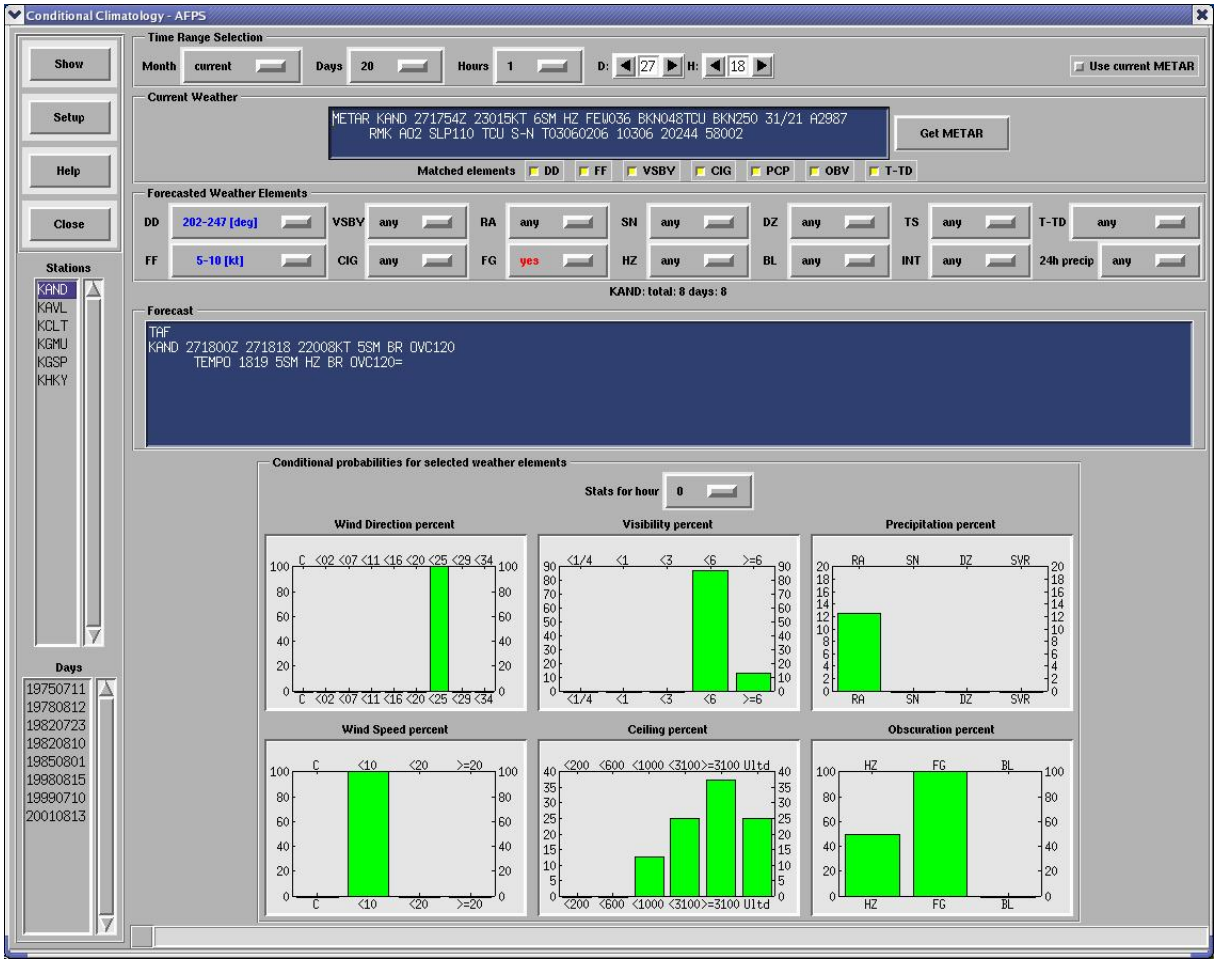
To use the results of the search:

- A 6-h TAF forecast will be displayed in the Forecast text window. No attempt is made to combine similar groups in this guidance TAF.
- The **Days** list will show dates for which there was a match. If you select an item from this list, a popup text window will display a list of reconstructed METAR reports for that day.
- The bottom part of the GUI consists of six histograms displaying probability distributions for the forecast hour selected by the **Stats for hour** button. You can quickly switch between forecast hours.

Note

The sum of all probabilities on a histogram may exceed 100 percent. The program counts a hit when particular weather element occured (precipitation, obscuration), or was within a certain range (wind direction and speed, visibility, ceiling). The denominator in the probabiity ratio is the number of days, and usually there will be some change in weather conditions within the 2 hour window used to collect matching events.

Below is an example of the Climatological Statistics Tool performing a selection-based search:



Climate Statistics - selected elements

Note that **Time Range Selection** group has different content. Like the METAR-based search, the menus for selection of the month and number of days are present. The idea is to choose some weather element values of which you are reasonably sure, then let the program to fill in the gaps and generate a full forecast for the time set in the **Time Range Selection** section. The guidance TAF will include occasional weather as appropriate.

To perform a selection-based search:

- De-select the **Use current METAR** button. If it was on, the contents of the **Time Range Selection** section will change.
- Select a staion from the **Stations** list. The contents of the **Current Weather** section do not matter.
- Set the time range values for your search:
  - The **Month** button allows you to specify the month used in the search. Generally, the current month is sufficient.
  - The **Days** button specifies a range of days that will be used when searching. In the example, the date is set to the 12th of November, and the day range is set to 20. That means the search will span 20 days either side of the 12th of November - a window of 41 days.
  - The **Hours** button specifies a range of hours that will be used when searching. In the example, the hour is set to 1400 UTC, and the hour range is set to 1. That means the search will span times from 1300 UTC to 1559

UTC.

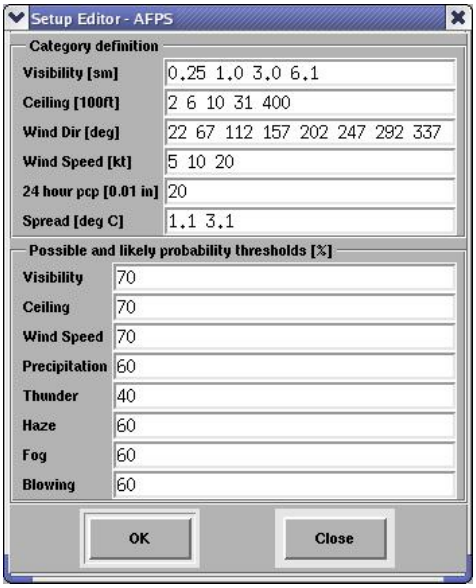
- The **D:** widget allows to set the day of the month.
- The **H:** widget allows you to set the hour of the day.

- Use the option menus in the **Forecasted Weather Elements** section to choose proper value(s). Avoid selecting too many elements or no matches will be found.
- Select the **Show** action button. The **Forecast**, **Days**, and **Conditional probabilities** sections will update after several seconds.

7.4. Climate Statistics Setup Editor

The Climatological Statistics Tool categorizes current and past weather for continuous elements to support its matching algorithms. You can configure the category boundaries via the Setup Editor. To launch the Setup Editor, select the **Setup** action button in the Climatological Statistics Tool.

Below is an example of the Setup Editor:



Climate Statistics Setup Editor

The Setup Editor has two sections, one for category definitions and the other for setting threshold probabilities.

Use the **Category Definition** portion of the Setup Editor to enter the boundary values. The entries are space-separated lists of values. Visibility and temperature spread (T - Td) are floating point numbers, all other elements must be integers. The entries must form an increasing sequence, possibly consisting of one element. Given a sequence

$$0 < V_1 < V_2 < ... < V_n,$$

the category k consists of values V such that

$$V_{k-1} <= V < V_k,\ k = 1,...,n+1 \text{ and } V_0 = 0,\ V_{n+1} = \infty.$$

Use judgement when configuring the values. Too many categories will lead to more unmatched events. Too few categories will yield a "fuzzy" forecast.

Use the threshold probability area of the Setup Editor to configure the tresholds that are used to select the most likely event from an ensemble (probablility distribution) of matching events. The algorithm used for finding the best value of wind speed, ceiling and visibility sums percentages from the lowest category to the highest category, until the sum of the probabilities exceeds the threshold probability. To determine occassional (i.e. TEMPO) conditions, the thresholds are multiplied by a factor of 0.7. That is, occasional conditions are always lower than prevailing.

8. New Climatology Tools

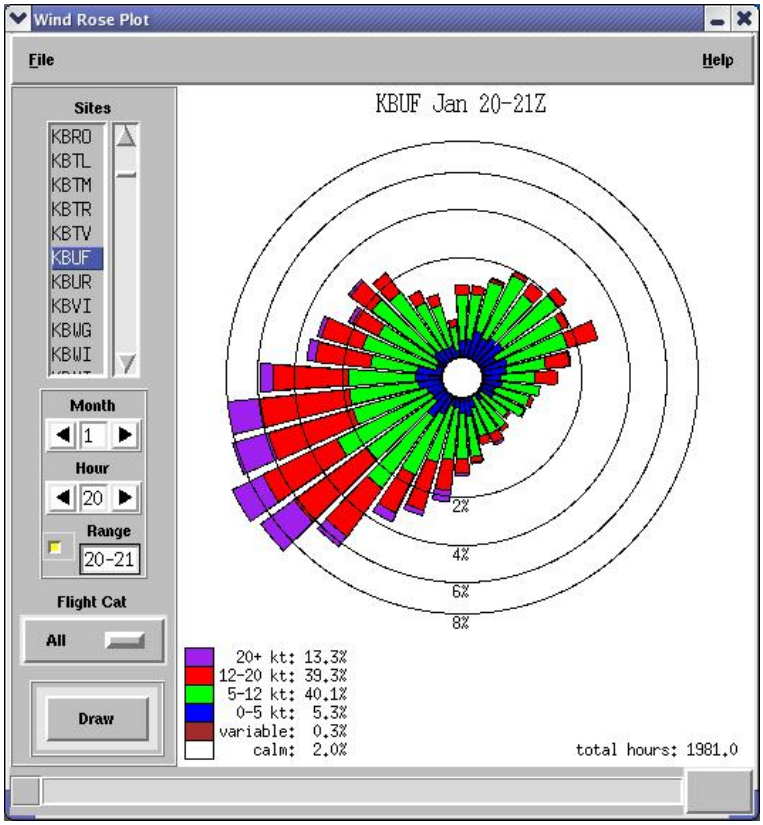
Climate tools introduced in AvnFPS 3.0 are being replaced. Release 3.2 preserves old GUIs described in the previous section. This section describes three new tools. However, these tools are not yet fully integrated with the rest of AvnFPS. In particular, a NFS access to climate data directory is required, however, that condition should be satisfied for many, if not all, AWIPS workstations. The new tools are started as standalone programs, from the AvnFPS Startup Menu, see Section 1, “Startup Menu” [].

Important

Observations in climate database have a notion of a “nominal” hour. The “nominal” hour is used for time range selection by the utilities described in this section. It is calculated as the full hour of observation time, offset by 10 minutes. That is, all observations with time HH:50 <= observation-time < HH+1:50 have their nominal hour set to HH+1.

8.1. Wind Rose

This GUI displays a wind rose for a given month and hour (or a range of hours). You can also specify flight category to further refine the results.



Wind Rose

The winds are displayed in a form of concentric wedges, split into segments reflecting wind speed. The area of each segment is proportional to the total time of wind falling into specific speed/direction category. It is possible to display wind statistics for observations with ceiling/visibility matching selected flight category. The “total hours” value, shown in the lower right corner of the display is calculated as follows:

- For a given month, all observations reporting wind are extracted from database, and sorted with respect to observation time. If **Flight Cat** menu is set to a specific flight category, only observations with this and lower categories are chosen for further analysis.
- For each pair of consecutive observations, it is assumed that the wind reported by the first one remains valid till the second one. To avoid gaps in data, a check is performed for the time interval between the observations. If the interval is greater than 1 hour 10 minutes, the pair is rejected. The time counter for the hours (1 - 3) falling within the calculated period is then incremented by appropriate value.
- Given selected hour, or hour range, the **total hours** display is the sum of all the time intervals that match the criteria specified above.

To display wind rose:

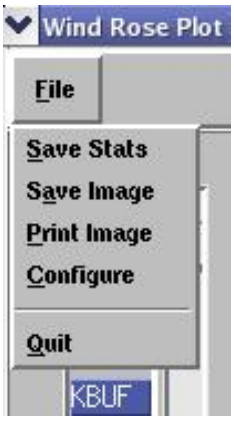
- Use arrows to set month in the **Month** counter.

- Use the arrows (or spinners) in the **Hour** counter to select an hour. If you want statistics for a range of hours, set the toggle button **Range** and type in the range of hours in the window as shown on the image above.
- Use the **Sites** list to see that site's data.
- Press the **Draw** button to draw the wind rose.

Note

Selecting a toggle or changing hour/month forces redraw.

The displayed data can be saved to a file as text or graphic and can be printed. Those features are accessible through the Menu Bar.



File Menu

To save statistics to a text file:

- Select the **Save Stats** menu item.
- Type the file name in the popup dialog, then press **Save**.

An example output file:

```
KBUF Jan
HOUR:          00Z
TOTAL:       991.8
CALM:         20.7
SPEED:       0-5 kt   5-12 kt  12-20 kt    20+ kt
VRB:           0.0       0.5       0.0       0.0
349-011:       5.5      16.2       1.5       0.0
011-034:       3.2      25.5       3.5       0.0
034-056:      10.7      32.2       3.9       0.4
056-079:      11.9      49.2      16.5       0.0
079-101:       1.5      21.8       4.0       0.0
101-124:       2.2      14.5       0.0       0.0
124-146:       4.0      17.0       2.1       0.0
146-169:       6.7      29.1       3.5       2.0
169-191:       3.3      35.2      11.1       1.0
191-214:       7.1      22.6      30.4       3.4
214-236:       3.4      42.4      52.7      19.1
236-259:       7.4      59.5      96.3      52.5
259-281:       3.6      41.1      68.6       9.2
281-304:       4.0      20.5      30.7       1.7
304-326:       3.5      16.0       4.1       2.1
326-349:       3.1      21.2       5.3       0.0

HOUR:          01Z
TOTAL:       989.8
CALM:         18.1
SPEED:       0-5 kt   5-12 kt  12-20 kt    20+ kt
```

```
VRB:            0.0       0.0       0.0       0.0
349-011:        3.8      14.6       3.0       0.0
011-034:        5.2      13.4       2.8       0.0

                 ...
```

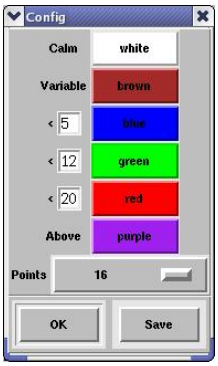
To save image to a file:

- Select the **Save Image** menu item.
- Type the file name in the popup dialog. To specify file format, use one of the extensions: .ps, .jpg, .png. Press **Save**.

To print image:

- Select the **Print Image** menu item.
- If needed, type in printer destination:
  `lpr -P printer`
  If the -P option is not specified, the image will be printed on your default printer.
- Press **OK**

To change wind speed thresholds and corresponding colors and to set number of wind directions use the **Configure** menu item. This will display a configuration dialog.



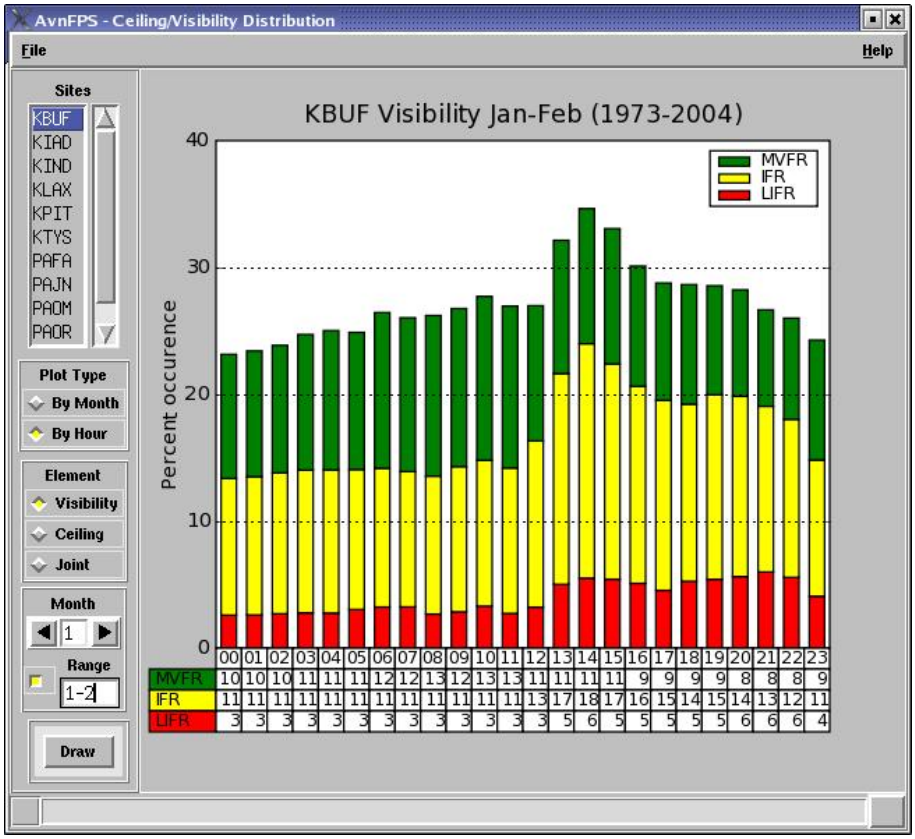
Configuration Dialog

- To change wind speed thresholds, enter appropriate values in the 3 available windows. Number of wind speed categories cannot be changed.
- To change colors, press a button to the right of the wind speed label/entry. A color editor will be shown.
- Use **Points** menu to select number of wind rose compass points, 8, 16 or 36.

Press **OK** to accept the changes. You may save the changes in a file, use **Save**.

8.2. Ceiling and Visibility Distribution

The Ceiling and Visibility Distribution GUI displays distribution of visibility and ceiling by month and hour. The display is in a form of a stacked histogram.



Visibility Distribution by Hour

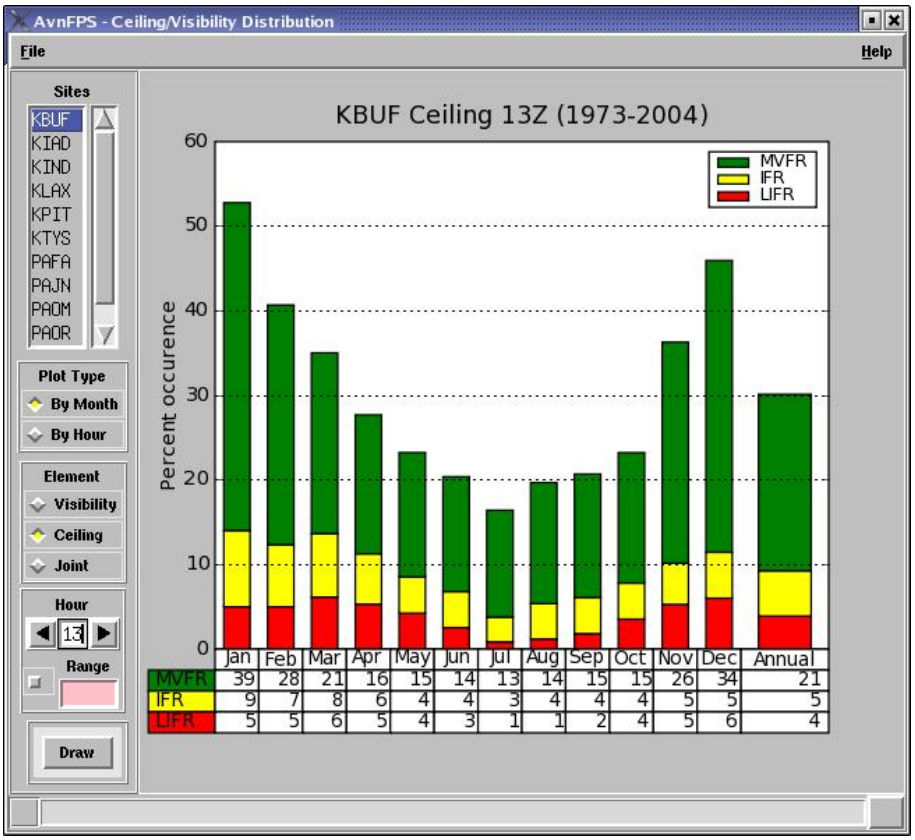
All ceiling and visibility data is extracted from the database. The timespan of a weather event is calculated similarily as in Wind Rose application. That is, observation are sorted with respect to time, the timespan of an observation is assumed to be the window between two consecutive observations, or 1 hour 10 minutes, whichever is less. The histogram and the table below displays relative time of occurence of weather resulting in all flight categories.

To display hourly distribution:

- In the **Plot Type** frame, select the **By Hour** toggle.
- In the **Element** frame, you can select the weather element to display by selecting radio button **Ceiling** or **Visibility**. If you want to see the flight category distribution, select **Joint** option.
- Use the arrows in the **Month** frame to select a month. If you want statistics for the range of months, set the **Range** toggle button and type in the range of months in the text window, as shown on the image above.
- Use the **Sites** list for the site of interest.
- Press the **Draw** button to draw the histogram.

To display hourly distribution, proceed as above, but select **By Month** toggle. Note that the selection area in the

lower left corner changes its label from **Month**to **Hour**. You can select either single or range of hours.



Ceiling Distribution by Month

Note

It may take up to a minute to retrieve statistics from climate data file. To speed up display, the toolkit uses a caching mechanism: the data is retrieved only once and then stored in a file, to be read on subsequent displays.

Note

Selecting a toggle or changing hour/month forces redraw.

The displayed data can be saved to a file as text or graphic and can be printed. Those features are accessible through the Menu Bar. The menu items are the same as for Wind Rose, with the exception of **Configure**. To save statistics to a text file:

- Select the **Save Stats** item.
- Type the file name in the popup dialog, then press **Save**.

An example output file:

```
KBUF (1973-2004)
HOUR: 00Z      Ceiling                 Visibility
MONTH     LIFR      IFR     MVFR       LIFR      IFR     MVFR      TOTAL
Jan      39.52    81.17   342.10      30.22   110.87    98.00     990.72
Feb      37.12    56.15   241.07      18.83    92.23    86.27     894.00
```

```
Mar     49.17   61.98  166.23      24.65   81.68   81.65     990.00
Apr     32.20   43.88  107.10       9.85   47.07   45.47     959.00
May     25.90   25.87   61.73       8.92   39.53   68.35     989.90
Jun      8.75   15.75   45.08       2.05   38.75   77.45     960.00
Jul      3.85    9.47   38.67       0.60   33.75   91.88     990.00
Aug      5.85   12.27   45.07       2.85   56.83  102.83     991.10
Sep     10.58   26.82   74.77       1.10   32.13   82.77     957.10
Oct     17.70   32.20  111.68       2.80   25.65   52.92     991.00
Nov     43.70   43.43  179.17      11.43   52.60   71.43     958.10
Dec     37.70   56.27  294.07      27.90   74.35   59.38     999.10

HOUR: 01Z      Ceiling                Visibility
MONTH    LIFR     IFR    MVFR       LIFR     IFR    MVFR      TOTAL
Jan     47.18   73.40  326.52      27.35  116.60   97.13     988.83

                           ...
```

To save image to a file:

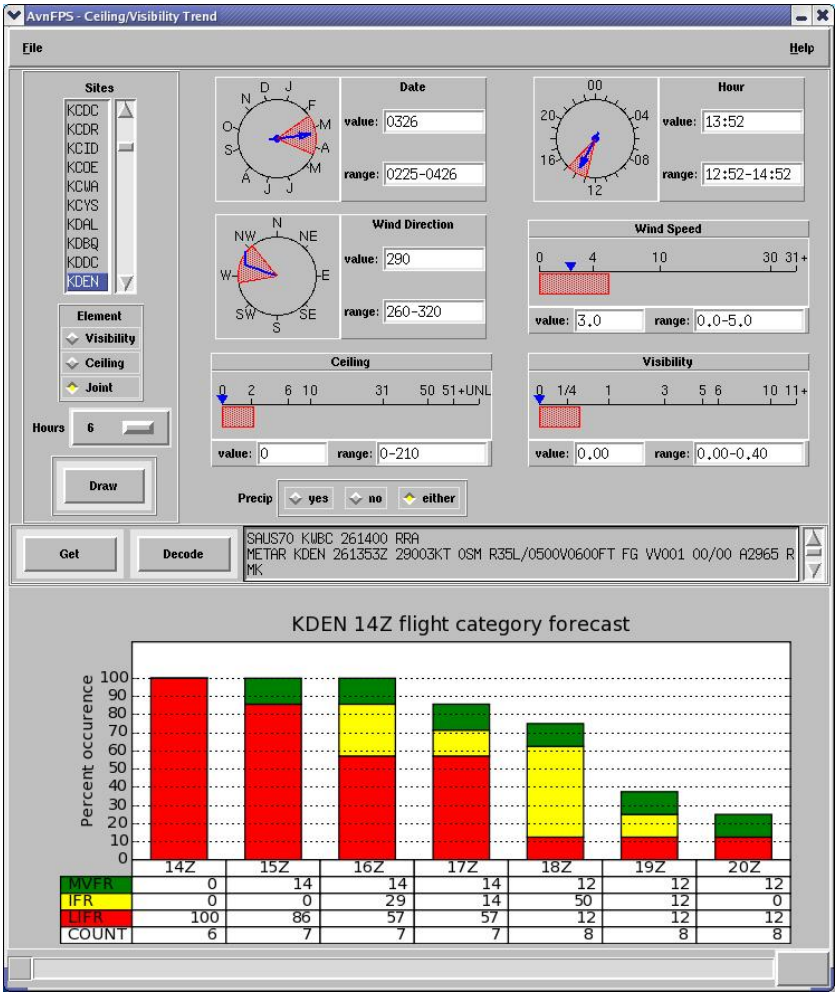
- Select the **Save Image** item.
- Type the file name in the popup dialog. To specify file format, use one of the extensions: .ps, .jpg, .png. Press **Save**.

To print image:

- Select the **Print Image** item.
- If needed, type in printer destination:
  `lpr -P printer`
  If the -P option is not specified, the image will be printed on your default printer.
- Press **OK**

8.3. Ceiling and Visibility Trend

The Ceiling and Visibility Trend GUI displays forecast of visibility and ceiling given initial conditions. The display is in a form of a histogram.



Visibility Trend

To display ceiling/visibility trend:

- Use the **Sites** list for the site of interest.
- Use the **Date**, **Hour**, **Wind Direction**, **Wind Speed**, **Ceiling**, **Visibility** and **Precip** widgets to set initial conditions. Each of the widgets contains a blue arrow indicating current value and a red strip/sector which is used to set matching criteria for searching similar events in climatological database. Left button moves value or an edge of range. Middle button is used to move both value and range. In the "Wind Direction" widget use right button to toggle between wind arrow and a circle representing calm or variable wind.

  Or, you can start with the most recent METAR. Select the **Get** button to retrieve the observation from the text database. The **Decode** button then initializes the selection widgets for you.

Note

The text window in which the METAR appears is editable.

- In the **Element** frame, select the weather element or, to display flight category, **Joint** toggle.
- Press the **Draw** button to draw the histogram.
- By default, a 6-hour forecast is displayed. You can change number of hours to 3, 6, or 12 from the **Hours** option menu.

The displayed data can be saved to a file as a graphic and/or can be printed out. Those features are accessible through the Menu Bar. To save image to a file:

- Select the **Save Image** item.
- Type the file name in the popup dialog. To specify file format, use one of the extensions: .ps, .jpg, .png. Press **Save**.

To print image:

- Select the **Print Image** item.
- If needed, type in printer destination:
  `lpr -P printer`
  If the -P option is not specified, the image will be printed on your default printer.
- Press **OK**

8.3.1. Algorithm Description

This tool searches climate database for events similar to those selected. If a matching event is found, the whole time period, i.e., user-specified time range, is marked as a match. In the second stage, for each matched period, observations with the nominal hours within the time range and number of hours selectable from the **Hours** menu are extracted. *Minimum* values of ceiling and visibility from observations for each nominal hour within the forecast range are used to determine flight category for the forecast hour. Relative frequency is displayed as a stacked histogram. The table below the histogram graphic shows percentage of occurence of each flight category.

Important

COUNT is the number of matching periods *NOT* the number of observations. COUNT will always be less than or equal of number of observations matching the selection criteria. Several matching observations within the same period do not increase the count. The reason for this is to not introduce bias in the calculations, as is typically the case when IFR conditions often result in increased observation frequency, compared to VFR (e.g. SKC) conditions.

The **Date** widget selects range of dates, lower and upper value inclusive. The month/day pair is converted to the day of the year. Leap years are not taken into account: If the selected date range consists of Feb 29 only, the search will use day 60 of the year, leap day or not.

The **Hour** widget allows resolution up to 1 minute. However the search is done for full hours. Lower and upper values of the selected range are converted to nominal hours. For example, range 1252-1452 sets selection limits to be 13-15, which will pick up observations within the time interval 1250-1550.

The **Ceiling** widget has to allow for an half-open interval (i.e. 5100 feet and up, sky cover over 50%) and unlimited ceiling. This requirement results in peculiar behavior of the red rectangle used to select range using the mouse. For example, if you want restrict ceilings to above 5000 feet, the rectangle should be reduced to a line at 51+.

Note

Database stores values in metric system (inherited from NCDC data files). The conversion between those and Imperial (US) units is never exact, consequently it is often impossible to guarantee selection of a single value, such as 3000 ft.

Appendix A. Editor Key Bindings

The editor is a standard Tk text window. Tk implements its own set of editing methods (called key bindings). The bindings below are quoted from [welch] [131], pp 385-387. The first 3 items on the list are specific to AvnFPS.

Table A.1. Key Bindings

Ctrl-u	Undo changes.
Ctrl-r	Redo changes.
Insert	Toggles insert/overwrite mode.
Any-Key	Insert normal printing characters.
Button1	Sets the insert point, clear the selection, set focus.
Ctrl-Button1	Set the insert point without affecting the selection.
Button1-Motion	Sweep out a selection from the insert point.
Double-Button1	Select the word under the mouse.
Tripple-Button1	Select the line under the mouse.
Shift-Button1	Adjust the end of selection closest to the mouse.
Shift-Button1-Motion	Continue to adjust the selection.
Button2	Paste the selection, or set the scrolling anchor.
Button2-Motion	Scroll the window.
Left or **Ctrl-b**	Move the cursor left one character. Clear selection.
Shift-Left	Move the cursor and extend the selection.
Ctrl-Left	Move the cursor by words. Clear the selection.
Ctrl-Shift-Left	Move the cursor by words. Extend the selection.
Right or **Ctrl-f**	Right bindings are analogous to Left bindings.
Alt-b or **Alt**	Same as **Ctrl-Left**, **Ctrl-Right**.
Up or **Ctrl-p**	Move the cursor up one line. Clear the selection.
Ctrl-Up	Move the cursor by paragraph which are group of lines separated by a blank line.
Ctrl-Shift-Up	Move the cursor by paragraph. Extend selection.
Down or **Ctrl-n**	All Down bindings are analogous to Up bindings.
PgUp, **PgDn**	Move the cursor by one screen. Clear the selection.
Shift-PgUp, **Shift-PgDn**	Move the cursor by one screen. Extend the selection.
Home or **Ctrl-a**	Move the cursor to line start. Clear the selection.
Shift-Home	Move the cursor to line start. Extend the selection.
End or **Ctrl-e**	Move the cursor to line end. Clear the selection.
Shift-End	Move the cursor to line end. Extend the selection.
Ctrl-Home	Move the cursor to the beginning of text. Clear the selection.
Ctrl-End	Move the cursor to the beginning of text. Extend the selection.
Ctrl-/	Select everything in the text widget.
Ctrl-	Clear the selection.
Delete	Delete the selection, if any. Otherwise delete the character to the right of the cursor.

Backspace or **Ctrl-h**	Delete the selection, if any. Otherwise delete the character to the left of the cursor.
Ctrl-d	Delete character to the right of the cursor.
Alt-d	Delete word to the right of the cursor.
Ctrl-k	Delete from cursor to the end of the line. If you are at the end of the line, delete the newline character.
Ctrl-o	Insert a newline but do not advance the cursor.
Alt-Delete	Delete the word to the left of the cursor.
Ctrl-t	Transpose the characters on either side of the cursor.

Appendix B. Monitoring Rules

Each of the monitoring modules has its own set of available rules. All rules will have the following characteristics: severity, type, unique toggle, and, optionally, message.

- Severity is the severity color which will be associated with this rule.
- Type is the name of the weather element which is monitored by this rule. With each type there is an associated label in the monitoring GUI. This is set in the configuration file `etc/gui.cfg` [13]. If a rule is used, its type must be listed on the items line.
- Unique is a toggle which determines whether the TAF Monitor will show two or more alerts of this type or limit its display to the most severe of the alerts. This feature is useful for rules that key on category differences. If one rule describes a difference of three categories as red while another rule describes a difference of two categories as yellow, situations that satisfy the red rule will always satisfy the yellow rule as well. If the Unique toggle is off, both rules will display in the TAF Monitor. If the Unique toggle is on, only the red rule will display. Message is a brief message that will be displayed in the TAF Monitor when the monitoring rule is triggered.
- Message, if set, is the text displayed in the popup window, when the rule evaluates to "True". If message is an empty string, it will be generated by the software.

This following tables describe available set of rules for each module.

Table B.1. METAR Monitoring Rules

Method:	DDDelta
Type: Unique: Args: Description:	wind 1 *DD*, *FF1* TAF and METAR wind directions differ by *DD* with either wind speed >= *FF1*.
Method:	FFDelta
Type: Unique: Args: Description:	wind 1 *FF*, *FF1* TAF and METAR wind speeds/gusts differ by *FF* with either wind speed >= *FF1*.
Method:	XFFMetar
Type: Unique: Args: Description:	wind 0 *RUNWAY*, *FF* METAR cross wind speed on runway *RUNWAY* exceeds *FF*.
Method:	CigCatDelta
Type: Unique: Args: Description:	sky 1 *NCAT* TAF and METAR ceilings differ by *NCAT* categories
Method:	CigTafThresh

Type:	sky
Unique:	0
Args:	*CIG1*, *CIG2*
Description:	TAF ceiling <= *CIG1* and METAR ceiling > *CIG2*.
Method:	CigMetarThresh
Type:	sky
Unique:	0
Args:	*CIG1*, *CIG2*
Description:	METAR ceiling <= *CIG1* and TAF ceiling > *CIG2*.
Method:	VsbyCatDelta
Type:	vsby
Unique:	1
Args:	*NCAT*
Description:	TAF and METAR visibilities differ by *NCAT* categories
Method:	VsbyTafThresh
Type:	vsby
Unique:	0
Args:	*VSBY1*, *VSBY2*
Description:	TAF visibility <= *VSBY1* and METAR visibility > *VSBY2*.
Method:	VsbyMetarThresh
Type:	vsby
Unique:	0
Args:	*VSBY1*, *VSBY2*
Description:	METAR visibility <= *VSBY1* and TAF visibility > *VSBY2*.
Method:	WxTafDelta
Type:	wx
Unique:	0
Args:	*WX* (a , separated list of weather elements)
Description:	Weather (any on the *WX* list) occurs in TAF and not in METAR
Method:	WxMetarDelta
Type:	wx
Unique:	0
Args:	*WX* (a , separated list of weather elements)
Description:	Weather (any on the *WX* list) occurs in METAR and not in TAF
Method:	WxMetar
Type:	wx
Unique:	0
Args:	*WX* (a , separated list of weather elements)
Description:	Weather (any on the *WX* list) occurs in METAR
Method:	WxVsbyDelta
Type:	wx
Unique:	0
Args:	*WX* (a , separated list of weather elements), *VSBY*

Description:	Weather (any on the *WX* list) occurs in METAR and not in TAF with observed visibility <= *VSBY*

AvnFPS 3.2 provides an experimental rule FltCatDelta. This rule is intended for use only at *authorized* Weather Forecast Offices. The rule monitors joint ceiling/visibility distribution. The intention is to limit number of alerts. Its behavior is different than the rules listed above: one rule covers all combinations of ceiling and visibility, its severity is generated dynamically. This contrasts with CigCatDelta and VisCatDelta where number of categories and severity are passed as arguments and the jusual setup is to have several instances of the same rule with different arguments. FltCatDelta is of type cat, takes no arguments, the message is generated dynamically. If used, number of threshold values for ceiling and visibility must be the same.

Table B.2. GFE Grid Monitoring Rules

Method:	DDDelta
Type:	wind
Unique:	1
Args:	*DD*, *FF1*
Description:	TAF and GRIDs; wind directions differ by *DD* with either wind speed >= *FF1*.
Method:	FFDelta
Type:	wind
Unique:	1
Args:	*FF*, *FF1*
Description:	TAF and GRIDs wind speeds/gusts differ by *FF* with either wind speed >= *FF1*.
Method:	CigCatDelta
Type:	sky
Unique:	1
Args:	*NCAT*
Description:	TAF and METAR ceilings differ by *NCAT* categories
Method:	SkyMismatch
Type:	sky
Unique:	1
Args:	*NCAT*
Description:	TAF and GRIDs sky covers differ by *NCAT* categories.
Method:	VsbyCatDelta
Type:	vsby
Unique:	1
Args:	*NCAT*
Description:	TAF and METAR visibilities differ by *NCAT* categories
Method:	WxTafDelta
Type:	wx
Unique:	0
Args:	*WX* (a , separated list of weather elements)
Description:	Weather (any on the *WX* list) occurs in TAF and not in GRIDs
Method:	WxGridsDelta
Type:	wx

Unique:	0
Args:	*WX* (a , separated list of weather elements)
Description:	Weather (any on the *WX* list) occurs in GRIDs and not in TAF

AvnFPS 3.2 introduced two additional rules: CigCatDelta and VsbyCatDelta. These rules are intended for WFOs that generate ceiling and visibilty grids with GFE.

Table B.3. Lightning monitoring rules

Method:	TsObsDelta
Type:	wx
Unique:	1
Args:	*NUM*
Description:	Number of reported lightning strikes within last 10 min >= *NUM*

Table B.4. Lightning probability forecast monitoring rules

Method:	TsInTaf
Type:	wx
Unique:	1
Args:	*PROB*
Description:	Thunder in TAF but guidance probability < *NUM*
Method:	TsNotInTaf
Type:	wx
Unique:	1
Args:	*PROB*
Description:	Thunder in not TAF but guidance probability >= *PROB*

Table B.5. CCFP monitoring rules

Method:	TsNotInTaf
Type:	wx
Unique:	1
Args:	*CONF*, *CVRG*
Description:	CCFP exceedes confidence and coverage thresholds with no TS in TAF.

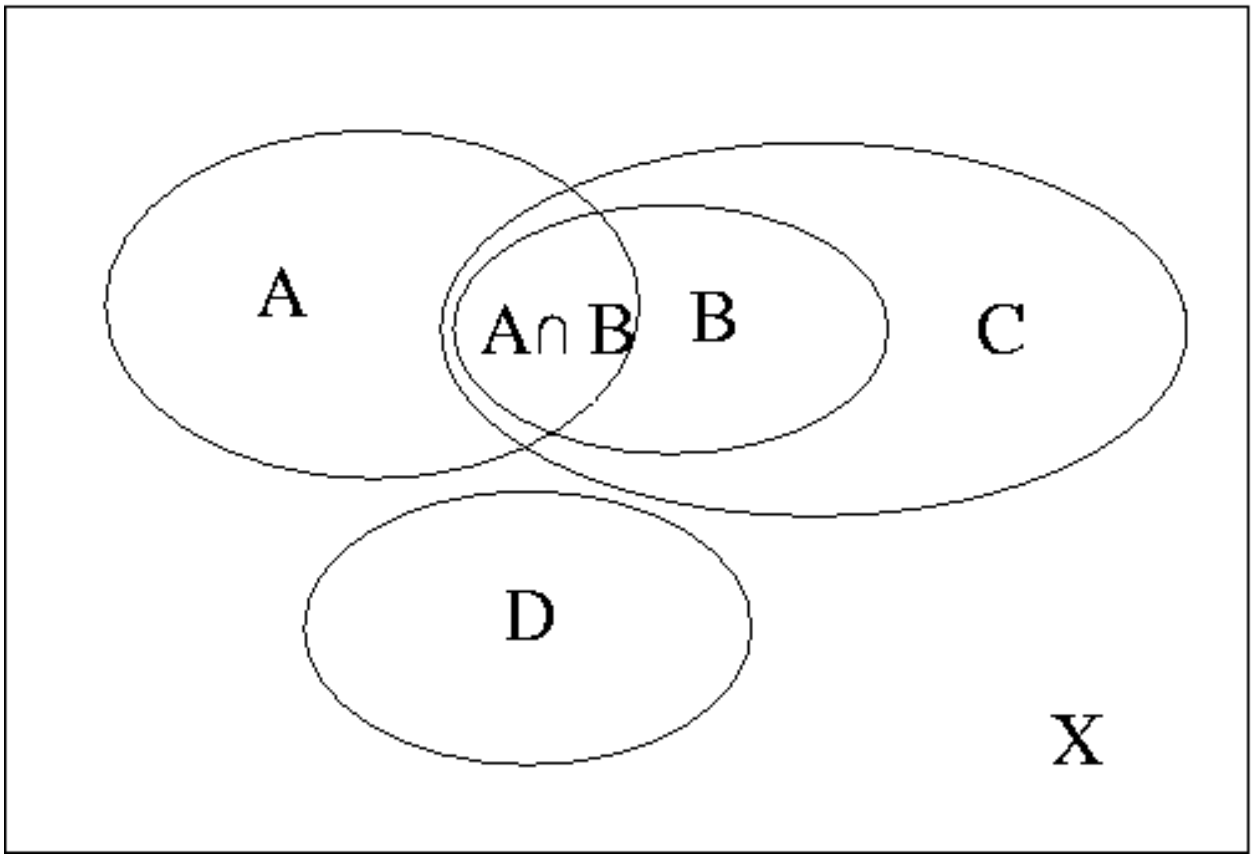
Table B.6. Low Level Wind Shear Monitoring Rules

Method:	WSinRadar
Type:	wx
Unique:	1

Args:	*VALUE* (1/s)
Description:	Derived wind shear value >= *VALUE*

Appendix C. Climate Quality Control

The algorithm used to check climatological consistency of a forecast is based on conditional probability estimates. The best way to illustrate the idea is to look at Venn's diagrams.



Venn diagram

An *event* is a subset of an *outcome space*, that is a set of all possible outcomes of an experiment. An event determined by random variable x is of the form $A = \{x \in A\}$. When the random variable is observed, that determines whether or not A occurs: if the value of x happens to be in A, A occurs, if not, A does not occur. Probability of an event (i.e. set) is its area, normalized such that the area of the outcome space X is one. Given two events A and B, the conditional probability of A, given B is defined by the formula

$$P(A \mid B) = P(A \cap B) / P(B)$$

Events A and B are said to be *independent* if

$$P(A \cap B) = P(A) * P(B) \Leftrightarrow P(A \mid B) = P(A)$$

Disjoint events, such as A and D on the above diagram, are *not* independent, they are exclusive: if D has happened, than A could not have happened. In this case

$$P(A \cap D) = 0 \Rightarrow P(A \mid D) = 0.$$

The other extreme case, where one set contains the other, illustrates dependent events: if B has happened, so must have C.

$$P(C \cap B) = P(B) \Rightarrow P(C \mid B) = 1.$$

These relations form the basis of our algorithm: we will claim that the combination of events is unlikely if

$$P(A \cap B) < \alpha_0 * P(A) * P(B)$$

or equvalently,

$$(*)\ \ P(A \mid B) < \alpha_0 * P(A)$$

where α_0 is a fixed arbitary value such that $0 < \alpha_0 < 1$.

The algorithm considers seven weather elements: ceiling, visibility, wind direction and speed, obstruction to vision, precipitation type and intensity. Obstruction to vision can be one of HZ, FG/BR, BLSN/BLSA or None. Precipitation type is one of SN, DZ, RA. Mixed precipitation is coerced to one of these types. Precipitation intensity is None, light, moderate or severe. Continuous elements are categorized: given a sequence of thresholds $0 < \tau_1 < \ldots < \tau_n$, the category k is a set

$$A_k = \{v: \tau_k <= v < \tau_{k+1}\},\ k = 0, ..., n+1,$$

with $\tau_0 = 0$ and $\tau_{n+1} = \infty$. Wind categories are a special case: instead of adding 0 and ∞ wind direction categories have to be "wrapped" modulo 360 degrees. An additional category is added for calm or light (that is, category 0) winds.

These categories form our outcome space, that is any weather instance can be qualified by a combination of the categories (i.e. events). Denote

$\{FF_k\}_{k=0,...n_1}$ - set of all wind speed events
$\{DD_k\}_{k=0,...n_2}$ - set of all wind direction events
$\{VSBY_k\}_{k=0,...n_3}$ - set of all visibility events
$\{CIG_k\}_{k=0,...n_4}$ - set of all ceiling events
$\{OBV_k\}_{k=0,...n_5}$ - set of all obstruction to vision events
$\{PTYP_k\}_{k=0,...n_6}$ - set of all precipitation type events
$\{PINT_k\}_{k=0,...n_7}$ - set of all precipitation intensity events
Let A_i be one of the events on the above list. Define $B_i = A_{k1} \cap ... \cap A_{k6}$, where k_j != i. Then $A_i \cap B_i = A_1 \cap ... \cap A_7$ By substituting these formulas into (*) we come up with 7 conditional probabilities.

$$(**)\ P(A_i \mid B_i) < \alpha_0 * P(A_i),\ i = 1, ..., 7.$$

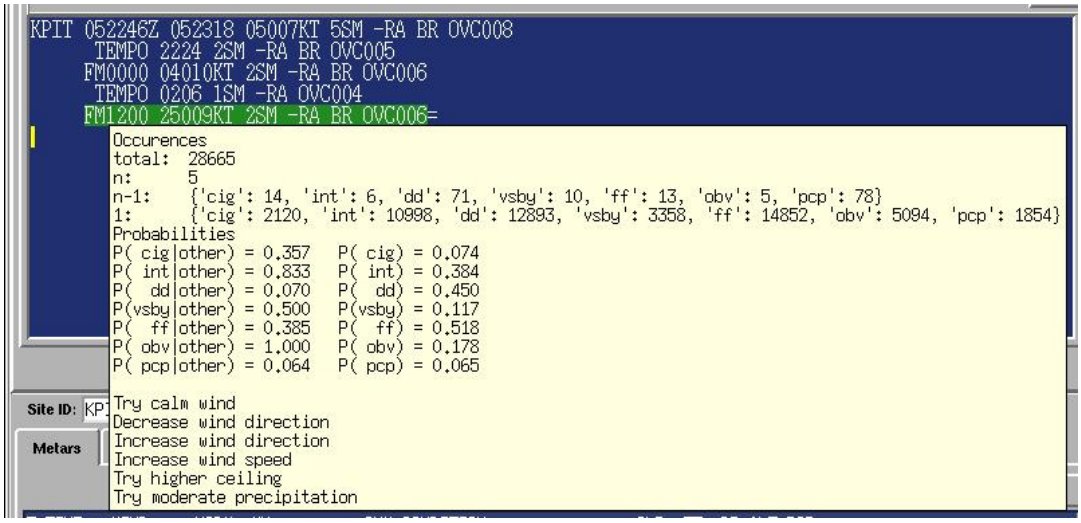
If the above inequality does not hold for some i we flag the corresponding weather element as suspicious.

The algorithm works as follows:

- Categorize weather in a TAF period.
- Evaluate minimal value α_{min} such that (**) is true for all i.
- If $\alpha_{min} \geq \alpha_0$, the forecast is plausible, otherwise continue.
- If $\alpha_{min} = 0$, none of the historical data matches forecasted weather. The software highlights the forecast in orange. If $0 < \alpha_{min} < \alpha_0$, the combination of weather elements is not likely and the forecast highlighted in green.
- An attempt is made to suggest more plausible weather: For each of the following elements: wind speed, direction, ceiling, visibility and precipitation intensity, a neighboring category is considered. The probabilities are calculated as above and a corresponding α_{min} value is calculated. If it is greater or equal to α_0, the weather combination is saved on a list.
- The list is sorted with respect to α and made available to the forecaster in the TAF Editor window.

The category thresholds and the value of α_0 are configurable, see System Administration, the section called “Climate QC Definition File” [20].

Here is an example of a detailed output:



Detailed Output

Appendix D. Low Level Wind Shear Monitoring

Low-Level Wind Shear (LLWS) monitoring has been implemented since AvnFPS 3.0. The WSR88D VWP tabular information and wind profiler data, along with surface observations at TAF sites allow vertical wind shear to be calculated and, with appropriate thresholds, alert the forecaster to the presence or the possibility of LLWS in the vicinity of their airports. As defined in [NWSI 10-813] [131], Section 1.2.8, LLWS is wind shear of 10 kts or more per 100ft in a layer more than 200 feet thick. Only winds within the lowest 2000 feet are used for LLWS calculations. LLWS monitoring alerts appear on the AvnWatch GUI, in a manner similar to the lightning alerts.

Important

Remember that vertical wind shear is not a scalar quantity, but a vector. Using just the difference in magnitude, i.e. speed shear, will, in the majority of cases, underestimate the amount of shear present. Direction of the horizontal winds must be considered as well.

On a benign weather day, wind shear values typically are less than 0.08 s^{-1}. Wind shear meeting official critera is 0.169 s^{-1}. The LLWS monitoring algorithm uses two threshold values which provides three wind shear regimes. These thresholds separates the wind shear spectrum into normal (green) and caution (yellow) and warning (red) regions. The placement of the thresholds is set via the AvnFPS configuration GUI and are customizable for each TAF site, see Section 4.1: “Editing Monitoring Rules” (page 34).

As you move your mouse over the LLWS alert light, the current TAF is displayed along with the source identifier producing the greatest shear and a "best-guess" (based on available data) wind shear group, suitable for placing into the TAF. The LLWS alert light will only change if elevated wind shears are detected and there is no WS group in your TAF forecast valid at that time. If there is a WS group in the TAF, the LLWS Alert light will *not* change, i.e. it will stay green, *regardless of the amount of wind shear detected*.

It is the WFO's aviation focal point's judgement to decide whether to use a radar's VWP product or profiler's data for a particular TAF site. Physical promixity and elevation of the data sources--radars and profilers--need to be considered as to whether LLWS monitoring is practical (or effective) for a particular TAF site. Placement of a radar on a mountaintop would be a disqualifying factor for use in LLWS monitoring since winds from such a location are likely to be unrepresentative of conditions over an airport that is situated at lower elevation. Keep in mind that wind shear impacting aircraft operations is limited to altitudes of 2000 ft AGL and lower. For a given TAF site, one can configure multiple upper-air sources to be used with the airport's surface observations. In such cases, the alert level on the AvnWatch GUI is based on the maximum shear found and is displayed to the forecaster.

To begin LLWS monitoring for your TAF sites, see Section 4.2: “Editing TAF Site Information” (page 37).

In the event that the LLWS monitoring icon goes yellow or red, the forecaster should consult the VWP product of the profiler or radar that triggered the alert.

Important

If LLWS alert is based on radar data, the forecaster should examine the radar's VWP *text* product, not the graphic that's displayed in D-2D. The radar VWP text product shows greater vertical resolution in the lowest 2000 feet than what is displayed in the graphic.

With these data, the forecaster can judge the validity of the wind data and susequently the LLWS alert. If there is corroborative evidence from other sources that strong, non-convective wind shears are occurring, then amending the TAF to put in a WS group should be considered.

Appendix E. Ceiling and Visibility Grid Monitoring

One of the new features implemented with AvnFPS 3.2, is the ability to monitor IFPS ceiling and visibility grids in the AvnWatch GUI. This is an experiment pioneered by the Charleston, WV (RLX) WFO with a few other eastern region WFOs participating. The acquisition of such grids into GFE is described. However, the initialization, methodology or editing of such grids are not discussed here.

The required 'parms', predominate and categorical cloud heights and categories, and visibility grids, need to be added to the GFESuite ifpServer Fcst and Official databases. To do this, the GFESuite's localConfig.py file need the following entries:

```
CigHgt = ("CigHgt", SCALAR, "100ft", "Cloud Ceiling Height", 250.0, 0.0, 0, NO)
PredHgt = ("PredHgt", SCALAR, "100ft", "Predominant Cloud Height", 250.0, 0.0, 0, NO)
CigHgtCat = ("CigHgtCat", SCALAR, "index", "Cloud Ceiling Height Category", 6.0, 0.0, 0, NO)
PredHgtCat = ("PredHgtCat", SCALAR, "index", "Predominant Cloud Height Category", 6.0, 0.0, 0, NO)
Vsby = ("Vsby", SCALAR, 'mi', 'Visibility', 10.0, 0.0, 2, NO)
```

And the following tuple added to the list 'parms':

```
([CigHgt, PredHgt, CigHgtCat, PredHgtCat, Vsby], TC1)
```

Once updated, the ifpServer will have to be stopped and restarted for these changes to take effect.

A GFESuite text formatter, `/awips/adapt/avnfps/3.2/py/IFPS2AvnFPS.py`, will need to be modified to query the new ifpServer parms and place them into the AvnFPS database. The octalthorpes, #, in the first column from lines 269 to 289 and lines 305 to 309 need to be removed. Save your changes.

To monitor these grids in AvnWatch GUI, new monitoring rules for grids need to be added. Using the the AvnFPS configuration tool 'Monitoring Critera' GUI, select the **grids** tab. A list of current monitoring rules for grids appear in the upper-half of the editor. In the available methods column, select the 'CigCatDelta' rule and then press **Add** button to monitor ceiling categories between your TAFs and the GFE grids. To monitor GFE visibility grids with your TAFs, select the 'VsbyCatDelta' monitoring rule and press the **Add** button. Press the **Save** to save your changes and restart the AvnWatch GUI.

Bibliography

[Aviation Workstation] Paul Kirkwood, David Hotz. “AWIPS Aviation Workstation Program Documentation”. Poster Session, 18th Int. Conference on Interactive Information and Processing Systems (IIPS) for Meteorology, Ocanography and Hydrology, Orlando, FL Amer. Meteor. Soc. 2002

[CCFP] *Collaborative Convective Forecast Product - Product Description Document*. URL: Aviation Weather Center [http://aviationweather.gov/products/ccfp]

[Kitzmiller et al.] David Kitzmiller. Mark Lilly. Scott Vibert. “The SCAN 0-3 Hour Rainfall and Lightning Forecast Algorithms”. URL: Unpublished Report [http://www.nws.noaa.gov/tdl/radar/03h_doc.htm]

[NDFD] Harry R. Glahn, David P. Ruth. “The New Digital Forecast Database of the National Weather Service” Bulletin of the American Meteorological Society: Vol. 84, No. 2, pp. 195-201 2003

[NWSI 10-805] “National Weather Service Instructions 10-805” Operations and Services, Aviation Weather Services URL: Transcribed Weather Broadcasts [http://nws.noaa.gov/directives/010/010.htm] February 2005

[NWSI 10-813] “National Weather Service Instructions 10-813” Operations and Services, Aviation Weather Services URL: Terminal Aerodrome Forecasts [http://nws.noaa.gov/directives/010/010.htm] February 2005

[Pyro] Irmen de Jong. *Python Remote Objects version 3.5*. Feb 2005 URL: pyro.sourceforge.net [http://pyro.sourceforge.net]

[Python] *Python 2.4.1*. March 2005, Python Software Foundation URL: www.python.org [http://www.python.org]

[TPG] Christophe Delord, *Toy Parser Generator 3.0.5*. April 2005, URL: christophe.delord.free.fr [http://christophe.delord.free.fr/en/tpg]

[Gamin] Daniel Veillard, *Gamin 0.1.7*. October 2006, URL: www.gnome.org/~velliard/gamin [http://www.gnome.org/~velliard/gamin]

[welch] Brent B. Welch. *Practical Programming in Tcl and Tk*. Second Edition. Prentice Hall PTR. 1997. ISBN: 0-13-616830-2